Sony Electronic Book Authoring System User's Guide

Creating Titles for an Electronic Book Player

August 1992

Sony Electronic Publishing Corporation
Publishers Data Service Corporation
One Lower Ragsdale Drive
Monterey, CA 93940

Copyright

Information in this document is subject to change without notice and does not represent a commitment on the part of Sony Electronic Publishing Corporation. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Sony Electronic Publishing Corporation.

Copyright © SONY Electronic Publishing Corporation, 1992. All rights reserved. Printed in the United States.

Trademarks

Data Discman is a trademark of Sony Electronic Publishing Corporation. Microsoft is a registered trademark and MS-DOS is a trademark of Microsoft Corporation.

Contents

Foreword
Audience Assumptions ix
Notational Conventions x
What's In This Guide xi

1 Introduction
About the Sony Data Discman 1-1
About the SEBAS Tools 1-3
About SGML 1-4
Producing an Electronic Book 1-4

2 Installing the Sony Electronic Book Authoring System
System Requirements 2-1
Installation Procedure 2-2
Contents of the Release Diskettes 2-5
Directory of \SEBAS\BIN 2-5
Directory of \SEBAS\EBXA 2-8
Directory of \SEBAS\TESTALL 2-9
Directory of \SEBAS\TESTSML 2-10
Using the Sample Electronic Books 2-11

3 Preparing Your Text
Preparing the Data Files 3-1
Deciding Which Search Functions to Use 3-2
Tagging Text With SGML 3-4
Summary of Tags 3-4
The DTD File 3-6
Word and Endword Search 3-6
Tagging the Headings 3-7

Making a Table of Contents 3-8
Creating a Table of Contents Manually 3-9
Creating a Table of Contents Automatically 3-11
Keyword Search 3-13
Making a STOPWORD File 3-13
Multi Search 3-15
Making the Question Files 3-15
Tagging the Fields 3-
Making a Reference Table Manually 3-17
Making a Reference Table Automatically 3-19
Formatting Text 3-20
Using References 3-21
About the Character Set 3-22
Inserting Consultation Links 3-24
Linking to Text 3-24
Linking to Graphics 3-25
Linking to Audio 3-26
Considerations 3-26
Checking for Syntax Errors 3-27

4 Preparing Your Graphics
Capturing Images 4-1
Converting Images 4-2
Inserting Consultation Links From Graphics 4-4
Linking To Text 4-5
Creating the Link Information File 4-5

5 Preparing Your Audio
About XA Audio 5-2
About Disc Format 5-3
Recording Audio 5-6
Interleaving the Audio Files 5-8
Interleaving with the SINTERLV Program 5-8
Interleaving with the INTERLV and MKSOUND Programs 5-9
Linking from Text 5-10
Creating the Link Information File 5-10
Check the Timings 5-12
Linking to Graphics with Audio 5-12
Creating the Link Information File 5-13

6 Building Your Electronic Book
Running the MKBATCH Program 6-1
The MKBATCH Questions 6-3
Running the BUILD.BAT Batch Program 6-12

7 Testing Your Electronic Book
Starting the Data Discman Emulator 7-1
Using the Emulator Keys 7-3

8 Preparing to Premaster
Premastering and Mastering 8-1
Setting Up the Files and Directory Structure 8-2
Making the Catalog File 8-3
Setting Up the Directory Structure 8-4
Testing the Directory Structure 8-5
Premastering and Mastering Using Sony CD-ROM XA Tools 8-6
Setting the Default Menu Language 8-8

9 SEBAS Command Reference
Notational Conventions 9-2
ADDSBENT 9-3
CDISO-DM 9-4
CDWO-DM 9-5
CHKSUB 9-6
DDMAN 9-7
DMPPCX 9-9
DMPVISTA 9-11
EB2PCX 9-12
EBIDENT 9-13
EXTWORD 9-15
FFFIXUP 9-17
FIF2EB 9-19
FONTS2EB 9-20
HDG2HSR 9-21
HSR2MHD 9-22
INTERLV 9-26
KSHEAD 9-27
KSINDEX 9-28

MHD2MSI 9-29
MHD2RMQ 9-30
MKBATCH 9-31
MKCATLOG 9-32
MKEMENUS 9-33
MKMAP 9-34
MKSOUND 9-35
MRGMENUS 9-
MSFIXUP 9-3
MSREFTAB 9-38
MSTABLE 9-39
NULLFILE 9-40
OPTSORT 9-42
PCX2EB 9-43
PCYEDIT 9-44
PCYHDR 9-46
PCYLEN 9-47
PCYPLAY 9-48
PCYREC 9-49
PCYSPLIT 9-50
PLAYER 9-51
PLAYXA 9-53
RECORDER 9-55
RERECORD 9-57
REVERSE 9-58
RPLSBENT 9-59
SGML2STF 9-
SHOWEBI 9-62
SINTERLV 9-63
STF2EB 9-64
VISTA2EB 9-66
WSHEAD 9-67
WSINDEX 9-68
Input and Output Files 9-69

A Creating an Electronic Slide Show
Creating the Graphics A-1
Converting the Graphics A-2

Preparing a Figure Input File A-2
Preparing an FIF File Automatically A-2
Preparing an FIF File Manually A-3
Building the Slide Show A-3
Testing Your Slide Show A-4

B Limitations and Issues
Results of the Keyword Search B-1

C Creating and Using Special Characters
Defining a Special Character C-1
Tagging Your SML File C-4
Creating Your Font Table C-4

D ISO 8859-1 Input Characters

E Getting Technical Support

Index

Foreword

The *Sony Electronic Book Authoring System User's Guide* is a set of instructions for publishers who want to create an electronic book for an electronic book player. This guide shows you how to use the Sony Electronic Book Authoring System (SEBAS) software tools from Sony Electronic Publishing Company to convert existing electronic data from any format to the electronic book format. The SEBAS tools run on a personal computer system under the MS-DOS operating system.

Sony Electronic Publishing Company, in cooperation with a variety of independent publishers, has used the SEBAS tools to publish many electronic books on such subjects as education, entertainment, literature, reference, sports, and travel and leisure.

An electronic book is ideally suited to reference material because the electronic book player has built-in software that searches for and retrieves data based on information you request using the keyboard.

Audience Assumptions

To use the SEBAS tools you need to know how to perform basic operations with the MS-DOS operating system or network operating system, and how to work with drives, directories, and files.

This guide assumes you are familiar with the operation of the Sony Data Discman electronic book player, or any functionally equivalent electronic book player.

There are several different tasks involved when creating an electronic book, and each task requires a different level of expertise.

The following table shows the skills that might be required for each task.

Task	Skills Needed
Deciding how to organize the data and what type of indices to produce.	Editor, book publisher, design engineer, user, professional indexer.
Tagging the data with the Standard Generalized Markup Language (SGML) tags for use with the SEBAS tools.	Word processor, or an engineer to write a program to automatically remove any unnecessary codes in the starting data and insert SGML tags into it.
Running the programs to build the electronic book.	DOS-literate person or software engineer who can understand the DOS Batch Command Language.
Testing the electronic book.	Individual to test the electronic book by running the Data Discman Emulator program on a personal computer.

Notational Conventions

The following notational conventions are used in this manual:

- `monospace font`

  You type exactly what appears in the format statement or the example as input, or the computer displays it as output.

- *`monospace font italics`*

You must supply a filename, value, or other parameter of the type specified. The _ underbar connects words in a parameter to make it more readable while showing it is all one item.

- UPPER-CASE CHARACTERS

  Actual filenames and filename extensions are shown in upper-case characters.

- [ ] Square brackets

  Square brackets surround optional items in the command line.

What's In This Guide

The *Sony Electronic Book Authoring System User's Guide* consists of nine chapters and some appendices, as follows.

Chapter 1, Introduction, tells you basic information about the Sony Data Discman, the SEBAS tools, the Standard Generalized Markup Language (SGML), and the production of an electronic book.

Chapter 2, Installing the Sony Electronic Book Authoring System, describes the system requirements for running the SEBAS tools, the installation procedure, and the contents of the release diskettes. It also shows you how to use the sample books.

Chapter 3, Preparing Your Text, tells you how to prepare and tag your text using SGML, how to set up the indices for the six electronic book search functions, format text, create text links, and check for syntax errors.

Chapter 4, Preparing Your Graphics, tells you how to prepare and convert graphic images and insert text and graphic links for the electronic book graphic search function.

Chapter 5, Prepaing Your Audio, provides an overview of how to capture, edit, and interleave your audio files. It also explains how to tie in your audio with text and images.

Chapter 6, Building Your Electronic Book, describes how to build the batch file required to process all of the files you make into one electronic book file.

Chapter 7, Testing Your Electronic Book, tells you how to use the Data Discman Emulator program on a personal computer, to test the operation of your electronic book as if it were running on an electronic book player.

Chapter 8, Preparing to Premaster, provides an overview of the CD-ROM premastering and mastering process, and shows you how to set the default language for your electronic book before you create your master on a CD-ROM write-once disc.

Chapter 9, Command Reference, details the command line syntax, options, and examples for each executable command in the SEBAS tool set, in alphabetical order.

The appendices provide information on creating an electronic slideshow from your graphic images, creating and using special characters, limitations and issues when making an electronic book, and how to get technical support.

NOTE: Sony Corporation requires data for the Sony Data Discman electronic book player to be in the mode 2 form 1 CD-ROM XA format, referred to in this guide as the CD-ROM XA format.

You can, however, test prototype databases on the Data Discman or on your personal computer using an ISO 9660 formatted disc.

1 Introduction

This chapter tells you about the Sony Data Discman Electronic Book Player, the Sony Electronic Book Authoring System (SEBAS) tools, the Standard Generalized Markup Language (SGML), and the basic steps for producing an electronic book.

About the Sony Data Discman

The Sony Data Discman Electronic Book Player is a small hand-held computer that reads a 3.2 inch (8 cm.) CD-ROM disc. The 3.2 inch CD-ROM disc holds 200 Megabytes of information. This is equivalent to 100,000 pages of printed text, 32,000 graphic images, or a combination of both. The disc can also store audio information. It is well suited to storing volumes of reference material.

The 3.4 inch diagonal liquid crystal screen displays 30 characters by 10 lines of text, and bit-mapped graphics at 256 x 200 pixel resolution. The typewriter-style alphanumeric keyboard has characters, a cursor pad, YES and NO command keys, and five function keys.

The main purpose of an electronic book player is to allow you to quickly access encyclopedic reference material, such as foreign language translations, city restaurant guides, dictionaries, anthologies, and so forth. The most efficient way to look up information in a traditional book is through the table of contents and the index. The same is true of an electronic book, which displays menus, or electronic tables of contents, on the screen. You choose a menu item to display the information listed under that topic. You can also type a word as if you were looking it up in an index, and the electronic book

player will find the information for you and display the associated text on the screen.

The electronic book software is a database search and retrieval system with six built-in functions, providing a variety of ways to access reference material.

Table 1-1 shows each built-in search and retrieval function and how it is used on an electronic book player.

Table 1-1. Search and Retrieval Functions

Search Function	Description
Word Search	Enter the first few letters of a topic word and the electronic book software finds and lists all topics that begin with those letters.
Endword Search	Enter the last few letters of a topic word and the electronic book software finds and lists all topics that end with those letters.
Keyword Search	Enter up to five words associated with your desired subject and the Data Discman finds and lists articles that relate to those words.
Menu Search	Scan Table of Contents menu to locate information.
Multi Search	Enter one-word answers to questions on the screen and the player finds the information based on your answers.
Graphic Search	A slide show style scan of images.

In addition to these six search functions, an electronic book player has a cross-reference capability that allows you to access related

information that has been linked to the topic or graphic you are viewing. These are called *consultation links*.

About the SEBAS Tools

The Sony Electronic Book Authoring System is a set of software tools developed by Sony Electronic Publishing Company for publishers who want to create electronic books for an electronic book player.

The SEBAS tools run on personal computers under the MS-DOS operating system. See "System Requirements" in Chapter 2 for specific hardware and software requirements.

The SEBAS tools consist of a set of executable programs, batch programs, and auxiliary files that help you convert your data into a format the electronic book player can use.

There are three general steps to converting data for use with SEBAS tools:

1. Make sure the starting data is stored on electronic media for access by personal computers running a version of the DOS operating system. If it only exists in book form, you will have to scan or type it into a personal computer system.

2. Strip out any existing typesetting or database codes until you have a plain text or ASCII file. The stripping can be done by a word processor or by a software program written specifically for the task. Because of the large volumes of data in reference material, typically a software engineer is assigned to write a program to filter the text.

3. Tag the text with the Standard Generalized Markup Language (SGML) tags so the SEBAS tools can process the data and create indices for the electronic book search functions.

About SGML

The Standard Generalized Markup Language (SGML) is a method for marking or *tagging* text for certain uses and for processing by various software programs. For example, one of the SGML tags is `<h1>`, which indicates that the text immediately following the tag is a first level heading.

SGML tags are used to identify heading levels for tables of contents and menu searches, words for word and endword searches, fields for multi search, and graphics for graphic search. SGML tags also identify links for consultation searches.

When you tag text for use with the SEBAS tools, the text must be in an ASCII file, a plain unformatted text file with no special characters, such as those used by word processor programs.

Sony Electronic Publishing Company uses a subset of the American Association of Publishers' subset of SGML.

By far, most of the time and effort in making an electronic book involves deciding what to index, then stripping out old codes or converting them to SGML codes, or just tagging plain text with the SGML tags. You can tag the data using a word processing program, or you can write a software program to tag the data.

Producing an Electronic Book

There are five general steps to creating an electronic book:

1. Start with an ASCII file of data that has been scanned or typed onto electronic media such as a hard disk. The data might contain typesetting or other database codes that need to be stripped out or converted into SEBAS codes manually by a word processor or automatically by a software program.

2. Tag the data for the different search functions that you want to support, using the SGML tags. Insert special text formatting instructions and links for text, graphics, and audio as appropriate.

3. Use the SEBAS software tools to build the indices for the search functions you are using and combine all the information into one electronic book file.

4. Test the operation of the book using the Data Discman Emulator on a personal computer.

5. Prepare the book for mastering onto a CD-ROM write-once disc using the CD-ROM XA tools.

This guide, the *Sony Electronic Book Authoring System User's Guide*, discusses steps 2, 3, 4, and 5 in detail.

2 Installing the Sony Electronic Book Authoring System

This chapter tells you what hardware and software you need to use the Sony Electronic Book Authoring System (SEBAS) tools, how to install the tools, and what is on the distribution diskettes.

System Requirements

The following hardware is the minimum required to run the SEBAS tools to create an electronic book:

- IBM 80386 or 80486 or compatible computer system.
- 2 Megabytes of Random Access Memory.
- 100 Megabytes of disk space for a typical 16 Megabyte database.
- An EGA or VGA monitor is required to run the Sony Data Discman Emulator program.

If you are using the audio portion of the SEBAS tools, you need the following hardware:

- Sony PCY-311 Audio board
- line-level audio source, (for example, a tape player) connected to the input jacks on the PCY-311 board

- headphones or amplified speakers connected to the PCY-311 output jacks

NOTE: The actual RAM and hard disk requirements depend on the database you are creating. In general, you need 1 Megabyte of RAM for each 30,000 consultation links.

Typically, during the build process, the required disk space is equal to 6 times the size of the database you are inputting. The resulting image file is usually 3 times the size of the input database.

The suggested RAM and disk space for running the SEBAS tools is 16 megabytes of RAM and a 600-megabyte hard disk.

The following software is required to run the SEBAS tools:

- MS-DOS or equivalent operating system
- QEMM386 or equivalent expanded memory system (EMS) driver. (QEMM386 is the only driver that has been tested thoroughly.)

Installation Procedure

SEBAS tools, version 2.02, is distributed on two disks each containing two files:

EBG/EBXA Tools Disk:

`install.bat`	the installation batch file
`sebas20a.zip`	the SEBAS tools packed into one file

XA/Audio Tools Disk:

`install.bat`	the installation batch file
`sebas20b.zip`	the audio and discman formatting tools packed into one file

NOTE: If you purchased the SEBAS tools without the audio license, you will have only an EBG/EBXA Tools Disk, and the sub-directory EBXA will not be present on your hard disk.

When you install the SEBAS tools, the software is copied from the floppy disk to the hard disk you specify, and the following directories are created on the hard disk:

`\SEBAS\BIN`	contains all the SEBAS executable files
`\SEBAS\TESTSML`	contains a simple SGML test file
`\SEBAS\TESTALL`	contains a more complex SGML test file
`\SEBAS\EBXA`	contains all audio and EBXA tools

Follow these steps to install the SEBAS tools onto your hard disk.

1. Place the EBG/EBXA tools disk into your floppy disk drive. **These instructions assume your floppy drive is drive B, and your hard disk is drive C. If your drives are different, substitute the appropriate drive letters.** Press the Enter key after each command line in these instructions.

2. Type the following command at the system prompt to make the floppy disk drive the default drive:

   ```
   C:>B:
   ```

3. Type the following command at the system prompt to copy SEBAS20a.ZIP to your hard disk, unpack the tools, and complete the installation.

   ```
   B:>INSTALL C:
   ```

   The screen displays this message:

```
Sebas Version 2.02 Installation
Install on C:
```

For a while you do not see any more messages on the screen, because the installation program is copying the SEBAS tools to the hard disk you specified.

Then a series of messages about unpacking the SEBAS tools appears. Finally, you see the message

```
Complete
```

4. Register your serial number. The DOSERIAL.EXE program runs and prompts you to enter your serial number. If you do not register your serial number the SEBAS tools do not create a valid electronic book, and you see the error message:

   ```
   *** Fatal Error *** Installation Failed
   ```

   If there is no error message then the installation is complete.

5. If you purchased the complete SEBAS package, place the XA/Audio tools disk into your floppy disk drive and type

   ```
   B:\INSTALL C:
   ```

   or the appropriate drive letter for your system. The directories described previously appear on the hard disk.

   NOTE: If you just type INSTALL and press enter, the command displays this message telling you the general form of the command:

   ```
   Sebas Version 2.02 Installation
   Usage:  install {drive:}
   ```

For instructions on running the test databases in TESTALL and TESTSML, see "Using the Sample Electronic Books" later in this chapter.

NOTE: You must add the SEBAS\BIN directory to your system path. If you are incorporating audio, you must also add the SEBAS\EBXA and PCY311 paths to your system path.

Contents of the Release Diskettes

This section lists the name and a brief description of each file that is copied to a subdirectory under the \SEBAS directory when you install the SEBAS tools. In the following list EB means "electronic book."

NOTE: The directories \SEBAS\TESTALL and \SEBAS\TESTSML contain test files with the same names. These files have slightly different contents, and are in their respective directories.

Directory of \SEBAS\BIN

The following list shows the files that are installed in the SEBAS\BIN directory, and provides a short description of each file. The notation EB stands for Electronic Book.

Filename	Description
ADDSBENT.EXE	Adds entry to the start block of a START file
BUILDALL.BAT	Batch program for more complex test file
BUILDSML.BAT	Batch program for simple test file
DDMAN.EB	Sign-on graphic in EB format for Data Discman Emulator
DDMAN.EXE	Data Discman Emulator for DOS computer
DMPPCX.EXE	Displays a PCX file as it would appear on an electronic book player
DMPVISTA.EXE	Displays a VISTA file as it would appear on an electronic book player

EB2PCX.EXE	Converts EB image to PCX file for editing
EBIDENT.EXE	Sets the default menu language after ISO formatting of electronic book file
EMPTY.DTD	Empty DTD satisfies SGML2STF warnings
EMPTY.REF	Empty REF satisfies SGML2STF warnings
EXTWORD.EXE	Extracts keywords from article for keyword search
FHFIXUP.EXE	Converts "fake" headings into actual headings
FIF2EB.EXE	Creates graphic search
FONTS2EB.EXE	Inputs special characters into .SML file
HDG2HSR.EXE	Produces a sorted heading file from a file of pseudo alphabetical headings
HLTSORT.CTL	Control file for OPTSORT program
HSR2MHD.EXE	Converts sorted heading file to menu search headings
ISO8859.ALT	Provides OPTSORT with sort order for ISO8859 character set
KSHEAD.EXE	Builds headings file for keyword search
KSINDEX.EXE	Builds index file for keyword search
LANGUAGE	Menu language file
MHD2MSI.EXE	Converts menu search headings to menu search index
MHD2RMQ.CTL	Control file for MHD2RMQ program
MHD2RMQ.EXE	Converts menu search headings to multi-search index
MKBATCH.EXE	Makes the batch program BUILD.BAT that builds the book into the *database*.BLD file
MKCATLOG.EXE	Makes the EB catalog file
MKEMENUS.EXE	Builds hierarchical browsing menu
MKMAP.EXE	Makes a map file
MRGMENUS.EXE	Merges hierarchical browsing menus

MSFIXUP.EXE	Fixes up all multi search tags in main text
MSREFTAB.EXE	Builds the multi search reference tables
MSTABLE.EXE	Builds the multi search question tables
NULLFILE.EXE	Creates a null length file
OPTSORT.CTL	Control file for OPTSORT program
OPTSORT.EXE	Sorts files (uses Opt-Tech sort program)
PCX2EB.EXE	Converts PCX file to EB image format
PURGE.BAT	Purges all non-source files for EB build
REVERSE.EXE	Reverses headings for endword search
RPLSBENT.EXE	Replaces a start block entry
SEBAS.DTD	Copied to PUBLIC.DTD for SGML2STF program
SEBAS.REF	Default REF file for SGML2STF
SEBAS.SYS	Copied to *database*.SYS for SGML2STF
SGML2STF.EXE	Checks the syntax for the SGML file and creates the SGML Token File *database*.STF
SHOWEBI.EXE	Displays one or more EB image files
SSF2SML.EXE	Converts Sony string file to SGML
STF2EB.EXE	Converts STF file to EB main article
STOPWORD	Default STOPWORD file for EXTWORD
VISTA2EB.EXE	Converts a VISTA file to EB image format
WSHEAD.EXE	Builds headings file for word and endword searches
WSINDEX.EXE	Builds index file for word and endword searches

Directory of \SEBAS\EBXA

The following list shows the files installed in the \SEBAS\EBXA directory, and provides a short description of each file.

Filename	Description
CDISO-DM.EXE	Discman EBXA formatter for image file creation
CDWO-DM.EXE	Masters image file to SONY CDW-E1/CDW-W1 unit
CHKSUB.EXE	Checks subheader information on audio interleaved file
DECODE.EXE	Plays back the audio you have encoded
ENCODE.EXE	Encodes your audio
INTERLV.EXE into	Interleaves different forms of XA audio one file
MKSOUND.EXE	Concatenates the interleaved files specified in the listfile.
PCY2XA.EXE	Converts C mono sound files into XA files
PCYEDIT.EXE	Trims the end of an audio segment
PCYHDR.EXE	Displays a sound file's header information
PCYLEN.EXE	Displays the length of a sound file
PCYPLAY.EXE	Plays a sound file
PCYREC.EXE	Records sound
PCYSPLIT.EXE	Splits one sound file into multiple sound files
PLAYER.EXE	Plays multiple sound files
PLAYXA.EXE	Plays an XA interleaved file
RECORDER.EXE	Records a sequence of sound files
RERECORD.EXE	Rerecords a group of sound files

SINTERLV.EXE	Interleaves C mono XA audio files
VOLUME.INF	Volume information file for Discman image

Directory of \SEBAS\TESTALL

The following list shows the files that are installed in the \SEBAS\TESTALL directory, and provides a short description of each file.

Filename	Description
LANK.EB	Sample EB format graphic
BLANK.PCX	PCX image file
DDMAN.EB	Sign-on graphic for DDMAN program
HLTSORT.CTL	Control file for OPTSORT program
IMAGINE.EB	Scanned calligraphy of John Lennon's "Imagine."
IMAGE.PCX	PCX image file
MHD2RMQ.CTL	Control file for OPTSORT program
OPTSORT.CTL	Control file for OPTSORT program
PUBLIC.DTD	Default definitions for SGML tags (used by SGML2STF)
PUBLIC.REF	Default REF file for SGML2STF
SEBAS.EB	Sony Electronic Publishing Company (SEPC) logo for initial screen in graphics search
SEBAS.FIF	Figure input file for SEBAS.EB (used by FIF2EB program)
STOPWORD	Sample STOPWORD file for EXTWORD
TEST.BLD	Sample electronic book file for use with Sony Data Discman Emulator
TEST.DTD	Empty DTD file satisfies SGML2STF warnings

TEST.I11	Reference table input for TEST database, multi search #1, question #1
TEST.MS1	Multi search table #1 input questions for TEST database
TEST.MST	Multi search table
TEST.R11	Reference table for multi search #1 and question #1
TEST.REF	Empty file to which PUBLIC.REF is copied. Satisfies SGML2STF warnings
TEST.SML	Sample SGML tagged ASCII text file
TEST.SYS	Satisfies SGML2STF
TESTFIG.EB	Test figure with graphic to graphic and graphic to text links
TESTFIG.IXY	Link information file for graphic to graphic and graphic to text links

NOTE: Other files in this subdirectory are created during the build process.

Directory of \SEBAS\TESTSML

The following list shows the files that are installed in the \SEBAS\TESTSML directory, and provides a short description of each file.

Filename	Description
BLANK.EB	Sample picture in EB format
BLANK.PCX	Sample picture in PCX format
DDMAN.EB	Sign-on screen for DDMAN program
HLTSORT.CTL	Control file for OPTSORT program
IMAGINE.EB	Scanned calligraphy of John Lennon's "Imagine" in EB format

IMAGINE.PCX	Scanned calligraphy of John Lennon's "Imagine" in PCX format
OPTSORT.CTL	Control file for OPTSORT program
PUBLIC.DTD	Default definitions for SGML tags (used by SGMS2STF program)
PUBLIC.REF	Default REF file for SGML2STF program
SEBAS.EB	Sony Electronic Publishing Company logo for initial screen in graphics search
SEBAS.FIF	Figure input file for SEBAS.EB (used by the FIF2EB program)
STOPWORD	Sample STOPWORD file for EXTWORD
TEST.BLD	Sample electronic book file for use with Sony Data Discman Emulator
TEST.DTD	Empty DTD file satisfies SGML2STF warnings
TEST.REF	Satisfies SGML2STF warnings
TEST.SML	Sample SGML tagged ASCII text file
TEST.SYS	Satisfies SGML2STF

NOTE: Other files in this subdirectory are created during the build process.

Using the Sample Electronic Books

The SEBAS tools are distributed with two small sample electronic book files. These files are included to give you a working example to study, change, and learn from. Each sample file is named TEST.BLD, but they are in separate directories as shown in the preceding section. The test book in the TESTALL directory contains a multi search index; the test book in the TESTSML directory does not. You can use the instructions in this section after you install the SEBAS tools.

You use the sample books with the Sony Data Discman Emulator program, DDMAN. To run DDMAN you must have an EGA or VGA monitor and graphics controller on your computer system.

Chapter 7, Testing Your Electronic Book, contains detailed instructions for using the Sony Data Discman Emulator program DDMAN. Refer to Chapter 7 for a list of keys to use with the Data Discman emulator. Otherwise, use the emulator exactly as you would use the Sony Data Discman.

NOTE: If your system has a mouse, you can use the mouse with DDMAN. Be sure to install your mouse driver (typically MOUSE.COM or MOUSE.SYS) before you invoke the DDMAN program.

You can type or print the ASCII files TEST.SML to see how the text has been tagged using the Standard Generalized Markup Language (SGML) tags.

To use the sample electronic books:

1. Be sure to include the \SEBAS\BIN directory in your system path.

2. Change the default directory to \SEBAS\TESTSML or \SEBAS\TESTALL, depending on which sample you want to run.

3. To invoke the Sony Data Discman Emulator with a sample electronic book, type the following command at the system prompt:

   ```
   DDMAN TEST.BLD
   ```

   The Data Discman Emulator sign-on screen appears stating the version number and copyright notice. Then the opening menu of the electronic book appears, showing the filename of the electronic book file TEST.BLD.

4. Press Enter or click on the YES key to display the main menu showing the six electronic book search functions.

5. Use the arrow keys or the mouse to select the menu function you want, then press the Enter key or click on the YES button to display the prompt for that search function.

6. Continue to use the Data Discman Emulator as you would use the Sony Data Discman electronic book player.

7. To exit the program and invoke another test book, type Ctrl-C, or use the mouse to click on the OFF button. DDMAN stops, the system prompt is redisplayed, and you are ready to change directories and start at step 3 again.

3 Preparing Your Text

This chapter tells you how to prepare your text for publication on an electronic book. It describes how to capture data to begin with, and what to do with it once you have it. This chapter tells you how to use Standard Generalized Markup Language (SGML) tags to tag your data for all of the electronic book player search functions, including making a table of contents and tagging for consultation searches. This chapter also shows you how to format text for a literal display, tag special characters, and check the resulting file for syntax errors.

Preparing the Data Files

Most of the time and effort to create an electronic book is spent preparing the data files for processing by the software that builds the search indices and formats the data for electronic book production.

You start with existing reference material, either in the form of hard copy pages, or electronic typesetting or word processing files. The existing reference material will be put on a CD-ROM disc where it can be accessed quickly in a variety of ways.

If the book is available in hard copy only, you can use an optical scanner to scan it onto electronic media such as a hard disk on a personal computer, then you must convert it to text with an optical character recognition (OCR) program. Or, you must type it onto electronic media using a word processing program.

Typeset files and most word processing files contain embedded codes that specify fonts, point sizes, boldface, italics, and other formatting information to the software programs that process them. It is

necessary to remove these codes from the files so they do not conflict with the codes you must enter for use with the Sony Electronic Book Authoring System (SEBAS) tools and the electronic book player.

If you must type or scan the text into a computer system, it is possible to enter it without any extra formatting codes. Word processors call this text only, plain text, or an ASCII (American Standard Code for Information Interchange) file.

The plain text will appear in one font and size, without any formatting other than carriage returns and automatic word wrap at a given margin. This means that you will have to look at the original hard copy to determine the breaks and divisions in the material that are usually indicated by titles, headings and subheadings. You need to know where these divisions are in order to set up the tables of contents menus, indices, and other search functions for the electronic book player.

Many companies hire a software engineer to write a program to remove the old formatting information from the data files, and replace it with the appropriate codes for use with the SEBAS tools and the electronic book player.

To do this, the engineer needs to know how you want to set up the data for display on the electronic book player, and the appropriate codes to do so. This chapter tells you how to set up text for the electronic book player. Chapter 4 tells you how to set up graphics. Chapter 6 tells you how to run the SEBAS tools software to put it all together.

Deciding Which Search Functions to Use

Deciding what to index, that is, telling the electronic book player which words it can find, is another difficult part of the process. It requires thoughtful consideration of each type of search function on the electronic book player, and its relation to the type of material you are referencing. Chapter 1 contains an overview of the search

functions of the electronic book player. Each search function requires that a certain type of index be built, or constructed, from the encoded text and the SEBAS *build* software.

Before you can completely decide, you need to know a little bit about tagging the text with the SGML tags, which is the coding system the SEBAS tools use to process text for the electronic book player. Generally, however, the following three search functions can be implemented from *one* SGML tag applied to the set of topics you want:

- Menu Search, displays a table of contents or topics from which you choose.
- Word Search, lets you enter the first few letters of a topic and displays all topics that begin with those letters.
- Endword Search, lets you enter the last few letters of a topic and displays all topics that end with those letters.

There is no extra tagging associated with a keyword search, which lets you enter up to five words associated with your desired subjects and displays articles that relate to those words. You just run the SEBAS build tools that make the index for keyword search. You should, however, indicate the words that you *do not* want to index, such as articles (a, an, the), prepositions, and other small words. This is done in the STOPWORD file described in the section "Keyword Search" later in this chapter.

Multi search is more complicated. You create up to five tables, each containing up to four questions that appear on the screen. The user provides one-word answers, and, based on the answers, finds the requested information. Multi search requires special tagging of each item to be accessed.

You can create links between text and graphics, text and text, text and audio, graphics and text, graphics and graphics, and graphics and audio. You must decide where you want the links, and then tag the target and the reference for each link. This is the consultation search

feature of the electronic book player. It is not accessible from a menu, but from within the display of text and graphics.

It is helpful to understand the structure of a book, its table of contents, headings, subheadings, body text, and index, to appropriately convert its parts for electronic searching.

Tagging Text With SGML

Tagging text generally requires that you insert a special combination of characters at the beginning of the text you want to tag, and another special combination at the end. This tells the SEBAS tools software where the text element begins and ends, and what to do with it.

NOTE: The filename of your tagged file should be the name of your database. The filename extension must be .SML. It is easier if the whole book is in one file when you are done tagging it, although you can specify multiple files when you process the tagged data. You should plan on running a small fragment through a test process to ensure your file is correct before processing a large file, and indeed, before completely tagging a large file. The fragment should be a complete representation of the actual source file. It should include all search types, images and audio links that the large version may contain.

Summary of Tags

Table 3-1 shows the SGML tags you can use in your data file.

The following sections tell you how to use the tags to format text and implement the various search functions on the electronic book player.

NOTE: The tags are not case-sensitive, but custom suggests using lower case, as shown in this table.

Table 3-1. SGML Tags

Tag	Meaning
<h1>	Begins first level heading; item appears in table of contents/menu.
</h1>	Ends preceding <h1> heading.
</>	Ends preceding tag.
<h2>	Second level heading.
	Begins new paragraph; terminates preceding tag.
	Ends preceding paragraph.
<!*text-of-comment*>	Comment, not included in book file.
<lit>	Begins literal text; prints exactly as shown, does not auto word-wrap lines, use to format tables.
</lit>	Ends literal text.
<ntr rid=*label-for-source*>	Begins note-in-text-reference-id, the source, the place from which you jump to a matching note-in-text target. End with </> or </ntr>.
<nit id=*label-for-target*>	Marks note-in-text-id, the place with the matching label that you jump to from the note-in-text-reference-id. Place at the beginning or end of the heading.
<fgr rid=*filename-of-figure*>	Begins figure-reference-id, the place from which you jump to the named figure, which is contained in the named file.
<snd rid=*filename-of-audio*>	Begins sound-reference-id, the place from which you jump to the named audio segment, whose audio play time is contained in the named file.
</fgr>	Ends figure-reference-id. Or use</>
Special Tags (not SGML)	
{MSFtQq}	Begins multi-search, where f=v for visible and I for invisible; t is the table number from 1 through 5; and q is the question number from 1 through 4.
{MS}	Ends multi search field.
{SCxx}	Special character tag; xx is a hexadecimal value of the ASCII code for the character
{ISOxx}	Tag for ISO 8859-1 character; xx is the hexadecimal value of the character.

The DTD File

The database.DTD file is the Document Type Definition file. It is used to define the SGML tags for the SGML2STF program. This file contains all the SGML tags that are shown in Table 3.1. If your text is already tagged you have two choices:

- You can remove the other tags and replace them with the SGML tags shown in Table 3.1.
- You can change the DTD file to give your tags the same meaning as the SGML tags.

The latter is much easier than the former, but can only be accomplished if the tags are unique and similar in structure. For example an SGML tag for a level one heading is <h1>. Your text might have <heading>. Instead of changing the tag, change the entry in the DTD file from h1 to heading. The SGML2STF parser will handle the data as a level one heading.

Word and Endword Search

Word and Endword Searches are based on tagged first level headings.

For Word Search, you enter the first few letters of a topic word, and the electronic book player finds and displays all the topics tagged as first level headings that begin with those letters.

For Endword Search, you enter the last few letters of a topic word, and the electronic book player finds and displays all the topics tagged as first level headings that end with those letters.

When you tag the headings, typically you label every heading as a first level heading so it will show up in Word and Endword searches.

This is a departure from normal book structure, but as you become familiar with electronic book production, you will see that it works.

Word and endword search allow you to do a partial or complete word or topic search (<h1> level) or a partial or complete search on the topic and the text in the document. Place tags inside angle (< >) brackets. Tags are processed by the SGML2STF program.

Tagging the Headings

There are three ways to tag headings. These examples show how to tag first level headings:

- You can explicitly terminate the heading tag by placing the </h1> tag at the end of the heading. This requires that you begin the next paragraph with the <p> tag, like this:

```
<h1>Document Heading</h1>
<p>This is the first paragraph of the topic.
The </h1> tag terminates the heading tag, and
the <p> paragraph tag begins the new paragraph.
If you do not begin the article with the <p>
tag, the syntax checker will display an error
stating that the text is untyped.
```

- Place an <h1> before the text, and leave a blank line after it, like this:

```
<h1>Document Heading

This is the first paragraph of the topic.  The
blank line above it acts as a tag terminator for
the <h1> tag in the heading, and then begins a
paragraph.
```

- Place an <h1> tag before the heading, do not leave a blank line after the heading, and begin the next paragraph with a <p> tag, like this:

```
<h1>Document Heading
<p>This is the first paragraph of the topic.
The <p> begin paragraph tag acts as a tag
terminator for the <h1> tag in the heading, and
then begins a paragraph.
```

NOTE: The most common SGML error is to forget to terminate a heading, either explicitly with a </h1> tag, or implicitly with a blank line following the heading and before the beginning of text, or implicitly with a <p> tag beginning the body of text.

Making a Table of Contents

There are two ways to make a table of contents with the SEBAS tools: automatically or manually.

The automatic table of contents can be used for databases that are hierarchical—that is, having multiple heading levels such as <h1>, <h2>, <h3>, and so forth. You can also use the automatic table of contents if the data consists of <h1> headings that are in alphabetical order. For all other table of contents structures, you must use the manual method.

A table of contents uses a consultation link from one spot in an electronic book to another—in this case, from a topic in a menu (table of contents) to the text that it references. In fact, on the electronic book player, a table of contents is a Menu Search.

NOTE: A table of contents entry (menu entry) is limited to 28 characters.

Creating a Table of Contents Manually

To create a manual table of contents, you must:

- manually enter and tag the topics that you want to appear in the menu on the electronic book player screen
- tag the text that goes with the topic.

When the user selects the topic from the menu, the associated text is displayed.

Typically, you add all of your table of contents entries to the beginning of the file.

Tagging the Heading Links

You can use a software program to do this, or you can manually copy all of the first level headings that you want to appear in a menu or table of contents to the top of the file.

Making the table of contents requires that you decide how to hierarchically structure how the user accesses the information in your electronic book. You make an entry that, when chosen by the user, leads to an article, which is typically another set of menus, until the type of information wanted is narrowed enough to display the text on the subject.

Here is an example of a simple table of contents for a small book.

To tag the heading links:

Start with `Table of Contents` as the very first line in the database. Label it as a first level heading. You do not need to link the first line because it is the top level menu record.

```
<h1>Table of Contents
```

Add the begin paragraph <p> tag, and the note-in-text-reference-id tag <ntr rid=*label-for-source*> to the beginning of the heading, and a terminator tag at the end of the heading, like this:

```
<h1>Table of Contents
<p><ntr rid=CHAP1>Chapter 1  Introduction</>
<p><ntr rid=CHAP2>Chapter 2  Installation</>
<p><ntr rid=CHAP3>Chapter 3  Preparing Text</>
```

Remember that each entry can contain up to 28 characters between the <ntr rid=*label-for-source*> tag and the </> ending tag.

Tagging the Text Links

Now you need to identify the text that each menu item displays. Add the matching note-in-text-id tag <nit id=*label-for-target*> after the text to display, like this:

```
<h1>Chapter 1  Introduction<nit id=CHAP1>

Welcome to the amazing world of electronic book
production.  In our example, this is the text in
the introduction in Chapter 1.  Note that you do
not need to terminate note-in-text-id tags.

<h1>Chapter 2  Installation<nit id=CHAP2>

This is the text under the heading
"Installation."  To install the software, turn
on, tune in, and drop everything.

<h1>Chapter 3  Preparing Text<nit id=CHAP3>

Preparing the text is the most difficult and
time-consuming part of preparing an electronic
book.  That is why you must hire an engineer to
do it for you.
```

You can also add the matching note-in-text-id tag *before* the text, like this:

```
<h1><nit id=CHAP1>Chapter 1  Introduction
```

Note that in a large electronic database, such as a guide to the culinary delights of the world, your first menu might look more like this:

```
<h1>Table of Contents
<p><ntr rid=AFRICA>African</>
<p><ntr rid=ASIA>Asian</>
<p><ntr rid=AUSTRALIA>Australian</>
<p><ntr rid=EUROPE>European</>
<p><ntr rid=NORTHAMER>North American</>
<p><ntr rid=SOUTHAMER>South American</>
```

Then the text you link to might be another menu containing the cuisines of major countries or groups of countries on each continent, such as Italian, French, and Portuguese under the European menu, and Japanese, Chinese, Thai, and Korean under the Asian menu. The next set of menus might contain a list of types of foods, such as vegetables, grains, fruits, seafood, meats, poultry, dairy, desserts, wines, and so forth. This continues until the user can arrive at something very specific to read about, such as a particular type of sushi.

Usually you can take a cue from the organization of the existing data. Sometimes you will have the same set of submenus under each of several main menus. Or, you might want to avoid this by reorganizing the data to have the duplicated menus be the main topics, with the different topics under them.

Creating a Table of Contents Automatically

If your data is tagged with at least first and second level headings, you can generate an automatic table of contents. The SEBAS tools do the following:

- Create a list of all the second level headings under each first level heading, all the third level headings under each second level heading, and so forth

- Automatically insert the note-in-text-reference-id <ntr rid=*label-for-source*> tags and the note-in-text-id <nit id=*label-for-target*> tags for each heading

The programs MKEMENUS and MRGMENUS create a hierarchical menu structure from the tagged headings. The <h2> level headings appear as links under the <h1> level, and the <h3> level headings appear as links under the <h2> level. When searching the document, the lowest heading level that contains a word is shown in the match list.

When you use MKEMENUS and MRGMENUS in conjunction with the FHFIXUP command and the -H2 switch with the STF2EB to create "file" headings, MKEMENUS and MRGMENUS create a hierarchical menu structure from the headings. The <h2> level headings appear as links under the <h1> level, and the <h3> level headings appear as links under the <h2> level. Run FHFIXUP at the end to "fix-up" the non-<h1> headings. When the user searches the document, the <h1> heading is shown in the match list even if the word is located in an <h2> or <h3> level heading. Refer to Chapter 9 for more information.

NOTE: When you run the MKBATCH program it asks you if you want to build a hierarchical table of contents. You answer *yes* to this question, and MKBATCH adds the proper commands to your BUILD.BAT batch file. The MKBATCH program is documented in Chapter 6, Building Your Electronic Book.

Making an Alphabetical Listing

When your data consists of first level headings that are in alphabetical order, you can generate a list of menus that are in alphabetical ranges. For example, if the data is a dictionary, you can use the automatic alphabetical table of contents feature to display words in the range A through D, words in the range E through H, and so forth.

You can use only <h1> headings for the alphabetical search. If the <h1> level is text it must be in alphabetical order. If the <h1> level is numeric it must be in numerical order. This option is provided for in the MKBATCH program. For numeric menus see the options on the HSR2MHD command in Chapter 9.

Keyword Search

The Keyword Search is the most automatic of all the electronic book player searches. There is nothing to tag, and nothing to label. The electronic book player searches and indexes the words in your database according to your answers to the questions when you run the MKBATCH program.

You can enter any word you can type on the keyboard into the Keyword Search function, and if it is in the book, the electronic book player will find and display it. Keyword search is designed to index the words in the document, but it can also index the topic (<h1>) level only, or both the topic and the document. Because Keyword search can index all words including single character words it is customary to prepare a list of words NOT to index, to save time and disc space. This list usually includes articles, such as a, an, and the, prepositions, and other small or frequently used connecting words.

The words you do not want indexed are listed in a text file called STOPWORD that the SEBAS tools access when building the electronic book. If a word occurs more than 1,500 times in a database it automatically becomes a stopword. This number can be increased to as much as 12,000, but the Discman only displays a match list of 400 words. Words that occur over 1,500 times should be indexed only if they are used in conjunction with other keywords.

Making a STOPWORD File

To make your own STOPWORD file, create a text file called STOPWORD. List each word to ignore starting in column 1 and

followed by a return. The entries in the STOPWORD file must be sorted in alphabetical order, and they must all be in one column.

Here are the words in our STOPWORD file.

```
A
AN
AND
ANY
ARE
AS
AT
BE
BUT
BY
FOR
FROM
HOW
IF
IN
INTO
IS
IT
ITS
OF
OR
SO
THAT
THE
THERE
THESE
THIS
THOSE
TO
WAS
WHAT
WHEN
WHERE
```

WHICH
WHO
WHY
WITH

Multi Search

Multi search allows the user to search for data based on answers to a series of questions created by the publisher. You, the publisher, must decide what questions to ask, and what the acceptable answers are.

You can create up to five multi search tables containing four questions each, for a total of twenty searchable fields per electronic book.

NOTE: If a word occurs more than 750 times in a database it automatically becomes a stopword. This number can be increased to 12,000, but the Discman only displays a match list of 400. Words that occur over 750 times should be indexed only if they are used in conjunction with other multi search questions in the same multi search table. See "Making a STOPWORD File" earlier in this chapter.

Making the Question Files

Name the question files as follows:

`DATABASE.MS1`	contains multi search table number 1 for the database
...	
...	
...	
`DATABASE.MS5`	contains multi search table number 5 for the database

Each question can contain up to 30 characters. Here is the content of a question file:

```
Number-of-questions-in-table
1st question (up to 30 characters)
2nd question
3rd question
4th question
```

Place an asterisk * at the beginning of the question to indicate that a reference table is available for this question. A reference table is a list of appropriate choices to help the user make a relevant selection. See "Making a Reference Table Manually" and "Making a Reference Table Automatically" later in this chapter for information about reference table files.

This is an example of a multi search question file:

```
4
*Last Name
*City
*Occupation
*Telephone number
```

Note that all of the questions have an associated reference table.

The electronic book player displays the following multi search screen for the example above:

```
Enter the words

Last name
[                    ]
City
[                    ]
Occupation
[                    ]
Telephone number
[                    ]
```

Tagging the Fields

There are two kinds of multi search fields:

- visible-text between the beginning and ending tags is shown on the electronic book player screen
- invisible-text between the beginning and ending tags is not shown on the electronic book player screen

Tag fields using the special multi search tag:

`{MSftQq}`*`text-being-tagged`*`{MS}`

where:

`f` = `V` for visible field or `I` for invisible field
`t` = multi search table number from `1` to `5`
`q` = question number from `1` to `4`
`{MS}` is the terminator for the multi search tag

The `{MSftQq}` and the `{MS}` do not appear on the electronic book player screen.

The electronic book player does not include `<h1>` headings in the multi search. In the following example, the `{MSI1Q1}` field is repeated at the bottom of the list as an invisible field, to enable the user to search on the name field. For invisible fields, the text between the beginning and ending tags is displayed as spaces, and should therefore be placed at the end of the article, preceding the next `<h1>`.

Here is an example of four tagged multi search fields in a file.

```
<h1>Discman, Data
<p>{MSV1Q2}Monterey, CA{MS}</p>
<p>{MSV1Q3}Manager{MS}</p>
<p>{MSV1Q4}(408) 372-7141{MS}</p>
<p>{MSI1Q1}Data Discman{MS}</p>
```

Making a Reference Table Manually

Each multi search field must have a reference table to provide suggested entries for the search field. The asterisk in front of the

question in the question file shows there is a reference table. The reference table for that question displays appropriate entries from which the user can choose. The table can also be set up like a menu search, to give the user broader categories from which to choose before he or she must make a specific choice.

The multi search reference table files must be named in the following format:

database.Itq

where:

database is the name of your database file
t = the multi search table number
q = the multi search question number

The contents of the multi search reference table file must be in the following format:

```
<h1>1st top level head
<h2>2nd level head, for 1st top level head
<h3>3rd level head, for 1st top level head
<h3>3rd level head, for 1st top level head
...
<h2>2nd level head, for 1st top level head
<h3>3rd level head, for 1st top level head
<h3>3rd level head, for 1st top level head
...
<h1>2nd top level head
<h2>2nd level head, for 2nd top level head
<h3>3rd level head, for 2nd top level head
<h3>3rd level head, for 2nd top level head
...
<h2>2nd level head, for 2nd top level head
<h3>3rd level head, for 2nd top level head
...
```

When the user selects an <h1> level heading, all of the <h2> headings under it are displayed; when the user selects an <h2> heading, all of the <h3> headings under it are displayed, until the user selects the bottom level. Then the electronic book player returns to the multi

search screen. The lowest level consists of the word that is tagged in the SML file.

The SEBAS tools currently support levels <h1>, <h2>, and <h3> only.

Here is an example of a multi search reference table file:

```
<h1>Cities A-D
<h2>Cities A
<h3>Anchorage
<h3>Atlanta
<h3>Auburn
...
<h2>Cities B
<h3>Birmingham
<h3>Boston
<h3>Burlington
...
<h1>Cities E-H
<h2>Cities E
<h3>Edgartown
<h3>Edinborough
<h3>Edmonton
...
```

Making a Reference Table Automatically

A reference table can automatically be generated by the SEBAS tools. The reference table is a 0 - 99999 and/or an A-Z hierarchy to arrive at the word or number that is tagged in the SML file.

The MKBATCH program contains an option to automatically create a reference table. You still need a `database.Itq` file, but it only needs to have one placeholder entry.

Here is an example of a placeholder `database.ill` file:

```
<h1>text
```

Formatting Text

The electronic book player automatically formats text to fit in the 30 column by 10 row screen display. In your electronic book file, you let text automatically wrap inside the margins of the file, until you need to separate a paragraph or a heading with a tag or a return character.

If you are displaying a table or other text that you want to keep in its existing format, without auto wraps, you can place a begin literal <lit> tag at the beginning of the text and an end literal </lit> tag at the end of the text. The electronic book player will display the text as shown. You can also use the <lit> tag to preserve white space.

The following example shows how to format literal text:

```
<lit>
Columns     of          Data
-----------------------------
Now         is          the
time        for         all
good        men         to
count       ...         0
1           2           3
4           5           6
7           8           9
</lit>
```

There is no horizontal scrolling on the electronic book player, so you must format your literal material to fit inside the 30 column screen display. If there is a consultation link within literal text then it must be formatted for a 28 instead of a 30 column screen. Two characters are needed for the link arrow. In literal text each complete consultation link must be on its own line. If an <ntr rid=label-for-source>text</ntr> tag wraps in the middle to the next line it is not a valid consultation link.

NOTE: We suggest you avoid using literal text as much as possible in electronic books for use on EB players that have

proportional fonts. SEBAS cannot determine the actual size of a screen when it uses proportional fonts.

Using References

You need to distinguish between characters used for the SGML tags and the same characters in your text. Specifically, the less than <, greater than >, and ampersand & signs in the text cause confusion and generate warning messages. These warning messages do not cause a problem with your data but they do fill your screen causing confusion and potentially hiding actual error messages. To avoid warning messages from the syntax checker, use the following substitute characters in the text:

Instead of	Use
&	`&`
<	`<`
>	`>`

The PUBLIC.REF file distributed with the SEBAS tools contains these codes plus codes that enable you to specify certain ASCII characters using the decimal equivalents 0 through 255. You can print the PUBLIC.REF file to view the codes.

Specify decimal equivalents of ASCII characters in the following format:

```
&#decimal-number;
```

For example, to preserve two spaces in front of a line, you can insert the decimal equivalent of a hard space at the beginning of the line. The following example shows two hard spaces at the beginning of a line:

```
<p>&#127;&#127;
```

About the Character Set

The electronic book player uses the Industry Standards Organization ISO 8859-1 character set for input and output of characters.

There are a total of 48 characters that you can input from the electronic book player keyboard. The electronic book player input characters are documented in Appendix D of this manual.

The electronic book player can display all of the defined standard ISO 8859-1 character set. Note that 7F hex is not defined, and 80H through 9F hex are invalid characters. The ISO 8859-1 character set is documented in Appendix D.

The SEBAS tools running on a personal computer expect characters to be in the standard IBM PC ASCII format.

The SEBAS tools translate the IBM character set into the ISO 8859-1 character set when you process your electronic book file. IS0 8859-1 characters are translated back into IBM PC ASCII for display by DDMAN, the Data Discman Emulator program.

Because of this translation process, all of the IBM PC ASCII line drawing characters are ignored because there are no ISO 8859-1 character equivalents.

The electronic book player can display some special characters, such as the accented e, (é) , that are in the ISO 8859-1 character set, but it does not include the accented e, (é) , on its keyboard.

If such characters appear in your electronic book, the SEBAS tools translate them from the IBM PC ASCII into the ISO 8859-1 character set for display in the main text data area, and into unaccented characters for use in the search indices. In this way, the user can type the unaccented character at the keyboard; the Discman can search on the unaccented character and find it in the index; and the accented character can be displayed on the screen.

NOTE: The SEBAS tools translate many of the accented characters in this way, but there are some that have no reasonable correlation in the ISO 8859-1 character set or the electronic book player input character set.

Tagging Special Characters

You can tag characters that are in the ISO 8859-1 character set and NOT in the IBM PC ASCII character set using the ISO tag in the following format:

```
{ISOxx}
```

where xx is the hexadecimal value of the character.

To tag a special ISO 8859-1 character, place the tag in the text where the character should appear, like this:

```
This sentence contains the character {ISOxx}
which is a member of the ISO 8859-1 character
set, but not a member of the standard IBM PC
ASCII character set.
```

NOTE: If the character does exist in the IBM keyboard, you can press the NumLock key to turn on the number pad, then press the Alt key, and then type the decimal equivalent of the character on the number pad. The character will appear on your screen, and the SEBAS tools will convert it appropriately.

Refer to Appendix C, Creating and Using Special Characters, for instructions on tagging special characters that you have created with the special character tag {SCxx}.

Inserting Consultation Links

The consultation search is different from the other search methods because the user chooses a consultation search while viewing an article or a figure.

The electronic book player indicates which words are linked to more information by displaying an arrow in front of the word or spot in the figure. When the user requests the search, the electronic book player displays the linked information.

When the user returns from a consultation link, the electronic book player displays the *beginning* of the article containing the link, not the position from which the link originated.

When the links are in a large article, the user must page down from the beginning of the document to the place from which he started, in order to continue reading the same text. Therefore, consultation links work best with small articles. Alternatively, you can place all the links at the beginning of each block of text.

Tagging text for a consultation search is just like tagging for a menu search or a table of contents, except that you do not have to create submenus to display.

Linking to Text

The source and target tags in the examples in this section are shown in bold to make them easier to read.

Place the note-in-text-reference-id tag <ntr rid=*label-for-source*> in front of the word to contain the link, and the </> terminator tag after the word, like this:

```
This is the text that contains a consultation
link to other text.  The link will be displayed
<ntr rid=textlink>here</> on the electronic book
player screen.
```

Place the note-in-text-id tag <nit id=*label-for-target*> at the beginning of the text to display, like this:

```
<h1><nit id=textlink>This is the text that will
be displayed on the Data Discmán screen when the
user chooses the word here from the previous
article.
```

The target of a consultation link must be a heading.

Linking to Graphics

Each graphic must exist in an individual file in the electronic book (EB) format on the disk. The graphics must be in the same directory as your SGML tagged text file. See Chapter 4, Preparing Your Graphics, for instructions on how to convert graphics.

Place the figure-reference-id tag <fgr rid=*filename-of-graphic*> in front of the word where you want the link to begin, and the </> terminator tag after the word, like this:

```
This is text from which we are making a
consultation link to a graphic.  In this case,
we place the link on the word that mentions the
<fgr rid=PICTURE.EB>picture</>.

When the user chooses the consultation search
for picture, the graphic file PICTURE.EB is
displayed.
```

NOTE: SEBAS forces a new line after a figure link.

Linking to Audio

You can create a consultation link from text to audio. This procedure is similar to inserting a link from text to a graphic. You insert a tag in the text and create a link information file. See Chapter 5, Preparing Your Audio, for instructions on how to capture and convert audio.

Place the sound-reference-id tag <snd rid=*filename-of-audio*> in front of the word where you want the link to begin, and the </> terminator tag after the word, like this:

```
This is the text from which we are making a
consultation link to a sound file.  In this
case, we place the link on the word that
mentions the <snd rid=AUDIO.XA>sound.</>
```

When the user chooses the consultation search for Audio, the audio segment described in the information file AUDIO.XA is played.

Considerations

There is a speed issue to consider when using large numbers of either image or audio files in DOS. Too many files in a subdirectory can cause DOS to allocate an extra cluster. This means the build utilities will make two accesses per file. This can slow down the build process dramatically.

Normally SEBAS requires that all files be present in the same subdirectory for building your electronic book image. You can place the image and audio file in separate subdirectories and include the path to the file in the <fgr rid=*filename-of-figure*> tag and <snd rid=*filename-of-audio*> tag. It is recommended that there be no more than 240 files per subdirectory. If you have 800 image files, you can create four different subdirectories and place 200 files in each one. When you tag your data with <fgr rid=> tags

specify the path in the tag, for example, `<fgr rid = imgset1\figure.eb>`. This points to a subdirectory called imgset1 that contains the images in figure.eb.

Checking for Syntax Errors

Once you have finished tagging your *database*.SML file, you must use the SGML2STF program to check it for syntax errors. If the syntax checker reports errors, you must correct them and recheck the file before continuing.

SGML2STF uses several files for input and produces several files for output. Later, the output files are copied into your current directory by the BUILD.BAT file created when you run the MKBATCH program. The MKBATCH program is documented in Chapter 6, "Building Your Electronic Book."

To check for syntax errors:

Run the BUILD.BAT file and watch for the SGML2STF program. While SGML2STF is running, you can use the CTRL=Break key combination to stop it and correct any errors in the SML file. SGML2STF is documented in Chapter 9, "SEBAS Command Reference."

4 Preparing Your Graphics

This chapter provides information about capturing and creating graphic images for your electronic book. It tells you how to convert the graphic images from their standard personal computer formats to electronic book player format. The last part of the chapter tells you how to create consultation links from graphics to text and to other graphics.

The instructions in this section assume you know how to work with directories and that you have included the SEBAS\BIN directory in your system path, to ensure that your system can locate the SEBAS programs described in this chapter and the graphic files on which they operate.

If you are making consultation links from graphics to text or other graphics, read the section "Inserting Consultation Links From Graphics" in this chapter, before finalizing your graphics.

Capturing Images

The Sony Electronic Book Authoring System (SEBAS) tools convert images in the VISTA (TGA) or PCX file formats into the electronic book (EB) graphic format. Therefore, your graphic images must be created in, scanned into, or converted into one of those two formats.

Once your graphics are in one of these two formats, there are two programs you can run to see how your graphics will look on the electronic book player: DMPPCX and DMPVISTA, which are documented in Chapter 9, "SEBAS Command Reference."

You might need to edit your graphic, look at it, edit it again, and continue until you are satisfied with its appearance.

To examine a PCX graphic:

Type the following command at the system prompt:

```
DMPPCX filename.PCX
```

Select the Show option.

The graphic is displayed on the screen as it will appear on the electronic book player.

To examine a VISTA graphic:

Type the following command at the system prompt:

```
DMPVISTA filename.TGA
```

The graphic is displayed on the screen as it will appear on the electronic book player.

When you are satisfied with the appearance of your graphics, you are ready to convert your images.

Converting Images

You use the PCX2EB and VISTA2EB programs to convert your graphic images into electronic book format. These programs, including their available command line options, are documented in Chapter 9, "SEBAS Command Reference."

The PCX2EB program is intended to convert PCX compressed monochrome images of 256 x 200 pixels (the resolution of the Sony electronic book player screen) into the Electronic Book image format.

PCX2EB attempts to convert a wide variety of other PCX formats, such as 4 or 8-bit color and gray scale, but with limited success. For larger images, its algorithm is to simply drop some of the pixels until it can fit the image into 256 x 200 pixels. The VISTA2EB program uses dithering techniques and is much more successful at converting larger color images into monochrome images of 256 x 200 pixels.

NOTE: If you are scanning your images, you should scan for 256 x 200 pixels with two-color dithering or posterization.

You do not have to use the filename extension .EB for your converted output files, but it is used here for clarity and it is recommended for easy recognition.

To convert a PCX image to electronic book format:

Type the following command at the system prompt:

```
PCX2EB filename.PCX filename.EB
```

The PCX2EB program converts the image in *filename*.PCX to the same image in the electronic book format, in *filename*.EB, or whatever filename and extension you specify.

To convert a VISTA image to electronic book format:

Type the following command at the system prompt:

```
VISTA2EB filename.TGA filename.EB
```

The VISTA2EB program converts the image in *filename*.TGA to the same image in the electronic book format, in *filename*.EB, or whatever filename and extension you specify.

In some cases it might be necessary for you to process the images using several of the available command line options, making more than one file, in order to compare the results. For example, you might compare 50% luminance versus optimal luminance. You can then use

the SHOWEBI command to display the choices for each graphic and select the best file for the electronic book.

To show an electronic book image (EB) file:

Type the following command at the system prompt:

```
SHOWEBI filename.EB
```

The image is displayed on the screen.

You can use the EB2PCX program to convert electronic book images back to PCX images for editing. The EB2PCX command is documented in detail in Chapter 9, SEBAS Command Reference.

To convert from electronic book to PCX format:

Type the following command at the system prompt:

```
EB2PCX filename.EB filename.PCX
```

Inserting Consultation Links From Graphics

You can create a consultation link from a graphic to another graphic or from a graphic to a text article. This procedure is very much like inserting a link from text to text or from text to a graphic.

You can specify up to nine consultation links for each graphic image.

When you create a consultation link from a graphic, you must tell the electronic book player where to display the black arrow cursor. You do this by specifying the X and Y coordinates of the place in the graphic where you want the link cursor to be displayed.

You need to determine the coordinates of your link on the image. Most paint (pixel-based drawing) programs have an option to display

the current cursor position. This can be useful in determining the coordinates for your link.

- The X-coordinate of a link must be a multiple of 8 pixels, in the range 8 to 248, inclusive. The X-coordinate of a link can be 8, 16, 24, . . . 240, 248.

- The Y-coordinate can be in the range 4 to 196, inclusive. The Y-coordinate need not be a multiple of 8.

After you determine the coordinates, you will place them in a link information file for each graphic image. The link information file is documented later in this chapter.

Linking To Text

You must use an SGML tag to indicate the text to display when the user selects the link in the graphic. The tag must be on the *heading* line of the text to display. The heading can be <h1>, <h2>, and so forth.

Place an SGML note-in-text-id tag <nit id=*label-for target*> in the heading line of the text you want to target, like this:

```
<h1>Text Heading<nit id=texttarget>

This is the text under the heading that contains
the target for the graphic link. The label-for-
target is texttarget.
```

The source for the graphic link, whether it is going to text or to another graphic, is specified in the link information file.

Creating the Link Information File

The link information file contains the source and target information for a consultation link from a graphic to text or to another graphic.

The link information file identifies the X and Y coordinates of the consultation link from a figure, and indicates whether the link is going to a figure or to text. The link information file also tells the filename of the target picture, or the *label-for-target* for the target text. In this sense it differs from creating consultation links for text, where you place the labels right in the text, and there is no link information file.

NOTE: You are required to supply link information for each picture that links to text or to another image. If a picture does not have a link, you need not supply a link information file.

To create a link information file:

Create a text file and give it the filename of your picture file, and the file extension .IXY.

Add the link information file in the following format:

Column 1-3	Column 5-7	Column 9	Column 11
X coordinate	Y coordinate	F or T	*Filename-of-image* or *label-for-target*

- Begin in column 1, and enter the X-coordinate of the link, then the Y-coordinate.
- Enter F if the link is going to a FIGURE; enter T if the link is going to TEXT.
- If the link is to a figure, enter the filename of the figure. If the link is going to text, enter the *label-for-target* from the note-in-text-id tag you placed in the text.

Here is an example of a link information file called FIGURE1.IXY. It contains all of the positions in the picture called FIGURE1 where the electronic book player displays the arrow to show there is a link, and the names of the target graphics and text.

```
40  40 F FIGURE2.EB
40  96 F FIGURE3.EB
40 152 T TEXTTARGET
```

NOTE: To make a hidden arrow, use coordinates X - 248 and Y - 184.

5 Preparing Your Audio

This chapter provides information about recording, editing, and integrating audio support for your electronic book. It tells how to record the audio using Sony's PCY-311 ADPCM audio encoding board, how to combine all your sound files into one interleaved sound file, and how to create consultation links between text, graphics, and audio.

The instructions in this section assume you know how to work with directories and that you have included the SEBAS\BIN, SEBAS\EBXA, and \PCY311 directories in your system path. The following hardware must be connected to your computer system.

- Sony PCY-311 audio board installed in your computer
- line-level audio source (for example, a tape player) connected to the input jacks on the PCY-311 board
- headphones or amplified speakers connected to the PCY-311 output jacks

The following steps provide a general overview of the process of creating a sound file. These steps are discussed in detail later in this chapter.

1. Record the audio sound bits using the Sony PCY-311 audio encoding board, attachments, and software.
2. Optionally, you can trim or remove "dead" sound from either end of the recorded audio files using the PCYEDIT program.

3. Interleave the recorded audio files into one master sound file. During interleaving a cue file is created that contains the start and end times of each individual file in the master sound file.

4. Create your audio links from text to audio, or from text to a graphic with audio.

5. Before you create your CD-ROM XA formatted image, rename the master sound file to SOUND.

The following sections discuss XA audio, disc format, and interleaving.

About XA Audio

Most music compact discs are recorded in the format known as "Red Book audio." In this format, sound is sampled about 44,000 times per second (44 kilohertz), and a 16-bit description of that slice is recorded. The high sampling speed and the large amount of data that can be included in 16 bits provide the highest quality audio available in the consumer market today.

However, there is a price for this sound quality: it takes a lot of space. An 8 cm. disc, the size used in the Data Discman, can hold 18 minutes of Red Book audio. That's not much if you are making a pronouncing dictionary or a talking book.

The CD-ROM XA standard solves this problem by defining two other sound quality levels, Level B and Level C. (Level A is not defined in the XA specification, so it is not covered here.) Both levels B and C save space by using only 4 bits, instead of 16, to describe the sound. This immediately quadruples to 72 minutes the amount of audio you can store on a disc.

To use the space even more efficiently, the sound is sampled less often—about 37.8 kilohertz (khz) for Level B and 18.9 khz for Level C. Some quick math shows that there is now 84 minutes for Level B

and 168 minutes for Level C. These times can be doubled again by recording in mono instead of stereo, giving 168 minutes (2 hours 48 minutes) for Level B mono and 336 minutes (5 hours 36 minutes) for Level C mono. Actually, the final result turns out to be a few minutes more (for example, about 5 hours 45 minutes for Level C mono) since the 8 cm. discs can hold a few seconds over 18 minutes of Red Book audio.

NOTE: These times are audio capacities. If you put 5 hours 45 minutes of C mono audio on a disc, it will completely fill the disc, leaving no room for any other data.

Because almost all Discman discs include other data such as text, graphics, indexing and link information, the practical limit for audio is less than the maximum for audio alone. How much less than the maximum depends, of course, on the amount and type of other data.

You probably already understand that the different sound quality levels have an effect on sound quality. Level B is roughly equivalent to the sound quality on FM radio, and Level C is about the same as AM radio, without the static. In other words, Level B is music-quality sound (though not as high quality as Red Book), and Level C is best suited for voice.

About Disc Format

All compact discs have the same basic layout. Data is inscribed in a continuous spiral from the hub out. This spiral is divided into *sectors*, the smallest unit that can be indexed. In Red Book audio format, each sector holds 1/75 second of audio; therefore, there are 75 sectors per second. Compact disc players, including the Data Discman and most CD-ROM readers now on the market, spin the disc at a speed that brings 75 sectors per second over the read laser.

Levels B and C audio, however, compress the audio data so there is more than 1/75 second of audio data in a sector. For example, C mono squeezes 16 times as much data into a sector; almost 1/5

second. Because the sectors are still passing over the laser at the same rate of 75 per second, this creates a problem: data is coming in faster than it can be played.

This concept is confusing, yet essential for preparing audio data for the Data Discman, so let's look at it from another angle. In Red Book audio, a sector passes over the laser, is read and played. This process takes 1/75 second. Because the disc is spinning at the rate of 75 sectors per second, everything works out fine: when the machine is ready for more data, the next sector in sequence is read. Data is read in at exactly the right rate, never running out (until the end of the disc), or delivering too much.

However, in C mono, there is enough data from one sector to play for nearly 1/5 second (16/75 second, exactly). By the time one sector of data finishes playing, 15 more sectors have passed over the laser—in other words, you can record only in every 16th sector. (In B stereo it is every 4th sector; in B mono and C stereo, every 8th sector.) It wastes disc space to use only 1/16th of the available sectors, so it is necessary to find a way around this problem.

The solution is *interleaving*, a technique used to record up to 16 audio *channels* together. Let's take a look at an example, again using C mono as our model. Let's assume that you are recording a book that has 16 chapters, one for each channel. You put Chapter 1, or *Channel* 1, in sectors 1, 17, 33, 49, and so on, in every 16th sector until you have recorded the entire chapter. Then you put Chapter 2 in sectors 2, 18, 34, 50, and so on—every 16th sector, starting in sector 2. Chapter 3 starts in sector 3, Chapter 4 in sector 4, and so on. When you have done this with all 16 chapters you have 16 interleaved channels. Then you tell the Data Discman which channel to read and which format the audio is in (so it knows how often to read a sector). The details of how to record and interleave audio files are discussed in the section "Interleaving the Audio Files" later in the this chapter.

Interleaving has an interesting effect on the capacity of a disc. The maximum of 5 hours 45 minutes can be reached only if the disc space is used 100% for audio, and 100% efficiently; in other words, there are no empty sectors. This turns out to be nearly impossible with

interleaving. You may already know why. Consider what happens if Chapter 1 is one minute longer than Chapter 2. Chapter 1 will continue to be recorded in every 16th sector, even after Chapter 2 is finished, so you will end up with one minute worth of blank sectors at the end of channel 2. When you multiply this inefficiency factor by 16, you can see that even a few seconds difference between channels quickly adds up. The amount of space your audio occupies is 16 times the longest channel, not the sum of the channels.

So far, the example used 16 channels. What if there are 15, or 17, or 5,000 channels? For less than 16 channels, it will still take 16 times the longest channel. There will be one entirely blank channel. For 17, or any number above 16, the audio will be divided into *segments*, with each segment holding up to 16 channels. Thus, if there are 17 channels, there is one segment with 16 channels and a second segment with 1 channel. If there are 5,000 channels, there are 312 segments of 16 channels each and one segment of 8 channels. The length of each segment is still 16 times its longest channel. The following figure shows how an audio file might be interleaved.

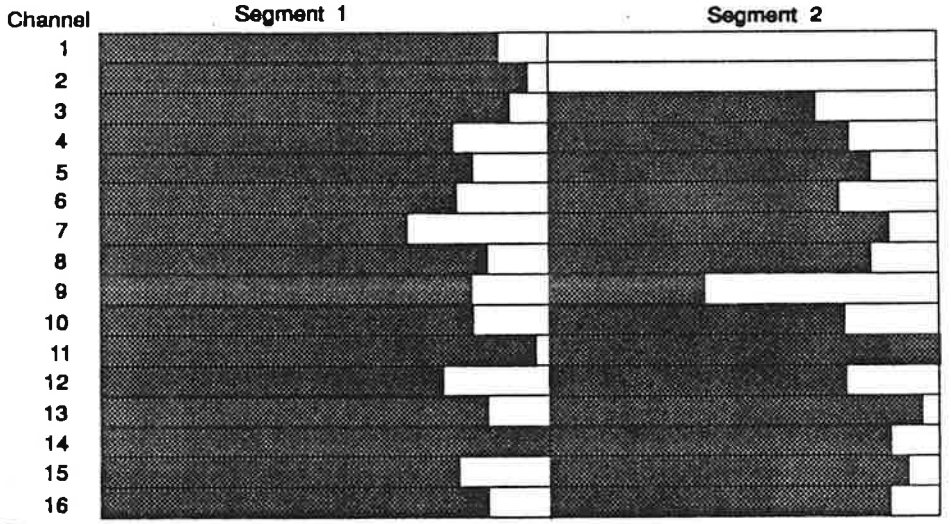

Figure 5-1. An Interleaved Audio File

Recording Audio

Your first task is to make a plan for recording your audio, based on the preceding information. Each channel is recorded into one file, and these files are then interleaved into one large sound file. Decide how many files (channels) you will record, and what you will record into each file. When you have done this, you are ready to turn on your computer and start encoding your audio.

You use the ENCODE and DECODE programs that come with your PCY-311 audio encoding board. Consult the documentation accompanying your PCY-311 board for more information on these programs.

To encode audio:

1. At the DOS prompt, change to the directory that will hold your audio files.

2. Type ENCODE and press Enter.

   - The program prompts you for an ADPCM file. Enter the filename to use for this audio segment.

   - The program prompts you for the level: choose BS, BM, CS, or CM, which stand for level B or C stereo or mono.

   - The program prompts you for the left and right attenuator values. You can experiment with these values to find the level that gives you the best quality recording; we recommend that you start with 0 and increase the levels as appropriate.

   - The program prompts you to start encoding. As soon as you press Enter the PCY-311 starts encoding the incoming audio. If you are working from a pre-recorded audio source, such as a tape deck, you might want to cue the tape in advance and

press the PLAY button on the tape deck at the same time as the Enter key on the computer.

- A stopwatch appears on the screen to time your audio recording. To use the "Mark" feature, press the spacebar at any time during encoding; the timing at that point appears on the screen while the stopwatch keeps running. Because of the PCY-311's buffer, this timing might be off by 3 or 4 seconds. Experience will tell you how much you need to add or subtract to compensate for the buffer. You can mark as often as you wish. Several marks will remain on-screen to provide a historical record of your marks in case, for example, you need to record the start and stop times of a section.

- Several other functions are available while you are encoding. You can adjust the attenuator levels up or down, pause (in case you want to change a tape), get the values in the command and status registers (useful mostly for programming the board), stop, or get help. The help option (-h) displays the list of available commands. This list also appears when the program first starts encoding.

- To play back the audio you have encoded, use the DECODE or PCYPLAY program. Each program plays back an ADPCM-encoded file recorded by the PCY-311 board. Run them without any parameters and follow the on-screen prompts. The PCYPLAY program is documented in Chapter 9.

At this point you might want to edit your sound files. For example, you can use the PCYEDIT program to remove dead space from the beginning or end of your sound files. The PCYEDIT program is documented in Chapter 9.

Interleaving the Audio Files

There are two ways to create the interleaved sound file.

- The first method, using the SINTERLV program, is only for audio encoded in the C mono format.
- The second method, using the INTERLV and MKSOUND programs, is used for B or C, stereo or mono.

The SINTERLV, INTERLV, and MKSOUND programs are documented in Chapter 9.

Interleaving with the SINTERLV Program

The SINTERLV program interleaves any number of C mono files, creating one sound file, a cue file with the extension .CUE, and a link information file for each channel. These link information files have the same name as the input filename, with the extension .XA. Link information files are discussed later in this chapter.

To interleave using the SINTERLV program, at the DOS prompt type

```
SINTERLV listfile output-filename
```

where

listfile is a plain ASCII file that lists, one on each line, all the files you want interleaved.

output-filename is the name of your output file.

Make sure that you have enough hard disk space to hold the resulting file, the cue file, and all the link information files. To figure the approximate length of your final file, multiply the total number of files by the length of your longest file. You can be more exact by

figuring the length of each segment and summing the size of all the segments. The simpler formula gives you a more conservative estimate that allows for a margin of error. The link information files are quite small—figure 256 bytes per file; there is a link information file for each of your original, pre-interleaved files. See the discussion under "About Disc Format" earlier in this chapter.

Interleaving with the INTERLV and MKSOUND Programs

The second method of creating an interleaved sound file might require several steps. The INTERLV program works exactly like the SINTERLV program above, except that it can only handle up to 16 audio files. If you have more than 16 files of B stereo or mono, or C stereo, you need to run the INTERLV program several times, each time producing a different output file, until all the files have been interleaved.

To interleave with the INTERLV program, at the DOS prompt type

```
INTERLV listfile output-filename
```

where

listfile is a plain ASCII file that lists, one on each line, all the files you want interleaved.

output-filename is the name of your output file.

Once this step is complete, you concatenate all these different segments together with the MKSOUND program.

To concatenate using the MKSOUND program, at the DOS prompt type

```
MKSOUND listfile output-filename
```

where

listfile is a plain ASCII file that lists, one on each line, all the files you want interleaved.

output-filename is the name of your output file.

INTERLV creates a single cue file with the name of the output file and the .CUE extension. MKSOUND produces one master cue file, with the name of the output file and the .CUE extension. This file contains the filenames of all the individual sound (.PCY) files, and the beginning and end times for each file. You can use these cue files to create your link information file as described below.

Linking from Text

You can create a consultation link from text to audio. This procedure is similar to inserting a link from text to a graphic. You insert an SGML tag in the text and create a link information file.

Place the sound-reference-id tag <snd rid=*filename-of-audio*> in front of the word where you want the link to begin, and the </> terminator tag after the word, like this:

```
This is the text from which we are making a
consultation link to a sound file.  In this
case, we place the link on the word that
mentions the <snd rid=AUDIO.XA>sound.</>.  When
the user chooses the consultation search for
sound, the audio segment described in the
information file AUDIO.XA is played.
```

Creating the Link Information File

The link information file contains the target information for a consultation link from text to audio.

If you are using C mono and interleaved with the SINTERLV program, this link information file already exists. If you want to play a section from beginning to end you can use this link information file. If, however, you wish to start somewhere other than the beginning or end somewhere other than the end, you need to create your own link information file.

If you interleaved with the INTERLV and MKSOUND programs, you need to create your own link information files using the information in the .CUE files created by these programs.

To create a link information file:

1. Create a text file and give it the filename of your audio file, with the file extension .XA.

2. Add the link information in the following format:

Column 1-9	Column 10-11	Column 13-20	Column 22
filename	type of audio	starting time	ending time

- ❑ Begin in column 1, and enter the name of the original, non-interleaved file.

- ❑ In column 10 enter 0 for B Mono, 1 for B Stereo, 4 for C Mono, and 5 for C Stereo.

- ❑ In column 13 enter the beginning time in the following format: minute:second:frame. *Frame* is the same as a CD-ROM *sector*, and is numbered from 1 to 75.

- ❑ In column 22 enter the ending time in the following format: minute:second:frame. *Frame* is the same as a CD-ROM *sector*, and is numbered from 1 to 75.

Here is an example of a link information file called CHAP1.XA.

CHAP1 04 00:36:50 00:37:71

Checking the Timings

Use the PLAYXA program to check the audio timings. If your sound file is called CHAP1.XA, at the DOS prompt type

PLAYXA CHAP1

You can use the function keys described on the screen to choose the channel, start, and end points to play. PLAYXA automatically chooses a valid frame for the channel you select. Be sure to check the frame (sector) number carefully; if you are off by one frame, you will be on a different channel. The PLAYXA program is documented in Chapter 9.

The EOR function toggles the End-of-Record switch. When EOR is on, play stops when the end of a segment is reached. When EOR is off, the audio ignores segment (record) boundaries and plays continuously until you stop (by pressing any key), or the end of the file is reached.

Linking to Graphics with Audio

You might want to display a graphic screen while your audio is playing. This is especially useful if you are creating a self-running slide show. The procedure for linking a graphic with audio is similar to that for linking to a graphic alone.

To link a graphic with audio:

1. First, create your graphic and convert it to the electronic book format. See Chapter 4, "Preparing Your Graphics."

2. Place the figure-reference-id tag that names the target graphic in your text. See "Linking to Graphics" in Chapter 3.

3. Create your link information file. Note there can be only *one* audio link in each link information file, there must be a link to another graphic, and there can be no links to text.

Creating the Link Information File

The link information file contains the target information for a consultation link from text to a graphic and from that graphic to audio.

The link information file identifies the X and Y coordinates of the consultation link from a figure to audio and to the next graphic, and indicates that the target of the link is audio. It also indicates the type of audio and the beginning and end times for the audio.

To create a link information file:

1. Create a text file and give it the filename of your picture file, with the file extension .IXY.

2. Add the link information in the following format:

Column 1-3	Column 5-7	Column 9	Column 11-12	Column 14	Column 23
X coordinate	Y coordinate	X	Audio type	Starting minute: second: frame	Ending minute: second: frame
X coordinate	Y coordinate	F	*Filename-of-image*.EB		

For both lines, enter the X and Y coordinates. Because this is an automatic link, you might want to place the arrow in the lower right corner of the picture, where it will not be so evident. To make a hidden arrow, use coordinate X-248 and Y-184.

In column 9, enter an X (for XA audio) on line 1, and an F (for figure) on line 2.

In column 11, line 1, enter 0 for B mono, 1 for B stereo, 4 for C mono, or 5 for C stereo.

In columns 14 and 23, line 1, enter the beginning and ending time in the following format: minute:second:frame. *Frame* here is the same as a CD-ROM *sector*.

In column 11, line 2, enter the filename of the next figure to display. This file has the extension .EB, and must have its own IXY file.

Here is an example of a link information file called FIGURE1.IXY.

```
248 184 X 04 07:33:59 09:58:32

248 192 F FIGURE2.EB
```

The result of this link information file is to play the specified audio segment while the graphic is on screen, then to automatically display the next figure, FIGURE2.

6 Building Your Electronic Book

This chapter tells you how to run the MKBATCH program. The MKBATCH program asks you questions, and according to your answers, creates a customized batch program called BUILD.BAT. You run BUILD.BAT to create your final electronic book file.

MKBATCH also creates a file containing your responses to the questions. You can use this file again for a different database, and it will use your previous responses as defaults.

Running the MKBATCH Program

To start the MKBATCH program, type the following command at the system prompt:

```
MKBATCH
```

The MKBATCH opening screen appears, and you are prompted to answer the first set of questions. After that, the screens you see depend upon your answers to previous questions. This section documents all of the questions you might encounter.

The default answers are shown in square brackets [ ].

NOTE: When in doubt, ALWAYS press the Return or Enter key to choose the default electronic book process. However, if you enter a different answer, your answer becomes the new default.

MKBATCH asks you to confirm your answers before it writes to the BUILD.BAT file or the response file.

- To accept the default shown in square brackets [ ], press the Return or Enter key.
- To enter an answer, type the answer, such as a filename, Y for yes, N for no, a requested number, and so forth, then press the Enter key. Y and N are not case-sensitive.
- To change your answers, press Enter when asked to confirm your choices for a given set. The default choice is NO. All of the questions are asked again, using your previous answers as defaults.
- To accept your answers for a given set, type Y and press Enter when asked to confirm your answers. The next set of questions is then displayed.
- To correct a mistake before you press Enter, press the Backspace key and type over again.

You can re-enter answers as many times as you wish for each set of questions, but once you accept a set of questions you cannot return to it. The information is not written to your BUILD.BAT file or your response file until the end of the process.

You can run MKBATCH again and re-create the BUILD.BAT file as many times as you wish. Then you can run BUILD.BAT to build the electronic book.

Experiment freely until you are satisfied, because the next step is making your CD-ROM master, and at that point it is more difficult to make changes.

The MKBATCH Questions

```
Response output file name [TEST.INP] ?
```

Press the Enter key to accept the default name of TEST.INP, or type a name for your response file, such as RESPONSE.FIL, and press Enter.

```
Output batch file name [BUILD.BAT] ?
```

Press the Enter key to use the name BUILD.BAT, or type your own batch file name and press Enter.

```
Directory containing SEBAS BIN files
[C:\SEBAS\BIN] ?
```

Press Enter to use the default path, or type a different path.

```
Database name [TEST] ?
```

Enter a name of up to 8 characters for your database, your electronic book title. Do not enter a file extension.

```
Does your database have images [N] ?
```

Press Enter if your database does not have images; type a Y and press Enter if it does have images.

```
Do you want menu search [N] ?
```

Press Enter if you did not create a table of contents or other type of menu hierarchy; type a Y and press Enter if you do want menu search.

```
Do you want keyword search [Y] ?
```

Press Enter if you want to keyword search; type N and press Enter if you do not want keyword search.

```
Do you want word search [Y] ?
```

Press Enter if you want word search; type N and press Enter if you do not want word search.

```
Do you want endword search [N] ?
```

Press Enter if you do not want endword search; type Y and press Enter if you do want endword search.

```
Do you want multi search [N] ?
```

Press Enter if you do not want multi search; type Y and press Enter if you do want multi search.

```
Redirect console output to log files [N] ?
```

This option allows you to send all of the console output from the BUILD.BAT batch program to log files, instead of seeing it on the screen. Press Enter if you do not want to redirect the output to files; type a Y and press Enter if you do want to redirect the output to files.

```
Delete all intermediate files (not recommended)
[N] ?
```

If you need to use the minimum amount of disk space, you can delete all of the intermediate files. However, you then cannot restart the build process without running BUILD.BAT from the beginning.

```
Confirm your above choices [N] ?
```

Press Enter to repeat all of the questions, allowing you to change any or all of your previous answers. Type Y and press Enter to accept your choices and continue on to the next screen.

NOTE: The answers in the square brackets are now your previous answers and are no longer necessarily the default.

Main Text Block Options

```
Enable browse from article to next article [N]
```

The browse function lets you page down from article to article, instead of stopping at the end of the current article. Press Enter to disallow browsing; type Y and press Enter to allow browsing.

```
Disable row of dashes between articles [N] ?
```

This question is displayed if you answer Yes to the previous question. The SEBAS tools can display a row of dashes between articles, so you can more easily see where one ends and another begins. Press Enter to display the row of dashes; type Y and press Enter if you do not want to display the row of dashes.

```
Do your figures have links to text (IXY files)
[N] ?
```

Press Enter if your figures do not have consultation links to text. If your figures have consultation links to text, you should have made link information files with the extension .IXY for each figure with a link. Link information files are documented in Chapter 4, Preparing Your Graphics. Type Y and press Enter if your figures have consultation links to text.

```
Do you want to indent the first line of
paragraphs [N] ?
```

Press Enter if you do not want to indent the first line of each paragraph. Type Y and press Enter if you do want to indent the first line of each paragraph.

```
Do you want automatic next/previous article
links [N] ?
```

The SEBAS tools can generate automatic consultation links to the next and previous articles. Press Enter if you do not want to create automatic next and previous article links. Type Y and press Enter if you do want them.

```
Do you want automatic outline links [N] ?
```

Press Enter if you do not want automatic outline links. Type Y and press Enter if you want automatic outline links.

```
Do you want a blank line after headings [N] ?
```

Press Enter if you do not want to display a blank line after each h1 heading. Type Y and press Enter if you do want to display a blank line after each heading.

```
Number of documents in your table of contents
[1] ?
```

Press Enter if there is one document (article) in your table of contents. Enter the number of articles in your table of contents if there is more than one. This could occur if you manually created your own hierarchical table of contents using more than one article. The number of documents is the actual number of <h1> tags needed to manually create your table of contents.

```
Confirm your above choices [N] ?
```

Press Enter to repeat all of the questions, allowing you to change any or all of your previous answers. Type Y and press Enter to accept your choices and continue on to the next screen.

NOTE: The answers in the square brackets are now your previous answers and are no longer necessarily the default.

Keyword Search Options

```
Do you want to search only headings [N] ?
```

Press Enter if you want to index words that occur in more than just headings; type Y and press Enter if you want to index words that occur in headings only.

```
Do you want to exclude headings from search [N]
?
```

Press Enter if you want to index words that occur in headings; type Y if you do not want to index words in headings.

```
Do you want to exclude numbers from search [N] ?
```

Press Enter if you want to index numbers; type Y and press Enter if you do not want to index numbers.

```
Do you want to include single character words
[N] ?
```

Press Enter if you do not want to index words consisting of 1 character only, such as A and I. Type Y and press Enter if you do want to index single character words.

```
Do you want to suppress "skipping" messages [N]?
```

If a word occurs more than 1500 times, the SEBAS tools do not index the word. When the SEBAS tools "skip" such a word, a message is displayed on your screen. Press Enter if you want to see the messages on the screen. Type Y and press Enter if you want to suppress the messages on the screen.

```
Confirm your above choices [N] ?
```

Press Enter to repeat all of the questions, allowing you to change any or all of your previous answers. Type Y and press Enter to accept your choices and continue on to the next screen.

NOTE: The answers in the square brackets are now your previous answers and are no longer necessarily the default.

Graphic Search Options

```
Do you want graphic search to appear in the mair
menu [Y] ?
```

Press Enter if you want a graphic search option to appear in the main menu. Type N and press Enter if you do not want a graphic search function in the main menu.

```
Do you have IXY files (for a slide show or link
from graphics) [N] ?
```

Press Enter if your figures do not want to use your consultation links to create an index of graphics. Instead, your figures will be accessed in sequential order by the graphic search function. Type Y and press Enter if you do want the graphic search function to use IXY files.

```
Do your IXY files contain links from graphics to
text [N] ?
```

Press Enter if you do not want to use your consultation links (IXY files) to text. If your figures have consultation links to figures, you should have made link information files with the extension .IXY. Link information files are documented in Chapter 4, Preparing Your Graphics. Type Y and press Enter if the figures have consultation links to text.

```
Do your IXY files contain links from graphics to
other graphics [N] ?
```

Press Enter if you do not want to use your consultation links (IXY files) to create a link to other graphics. If your figures have consultation links to figures, you should have made link information files with the extension .IXY. Link information files

are documented in Chapter 4, Preparing Your Graphics. Type Y and press Enter if the figures have consultation links to other figures.

```
Confirm your above choices [N] ?
```

Press Enter to repeat all of the questions, allowing you to change any or all of your previous answers. Type Y and press Enter to accept your choices and continue on to the next screen.

NOTE: The answers in the square brackets are now your previous answers and are no longer the default.

Menu Search Options

```
Is your first document a table of contents [N] ?
```

If you have made a table of contents, and therefore placed it at the beginning of the electronic book file, type Y to answer Yes to this question. If you have not made a table of contents, or if it is not your first article, press Enter to accept the default No.

```
Does your database contain hierarchical headings
[N] ?
```

If your database contains more than h1 headings, that is, also contains h2 and/or h3 headings, type Y to answer Yes to this question. If your database contains only h1 headings, press Enter to accept the default No. If you have hierarchical headings, the SEBAS tools generate an automatic table of contents. Refer to "Creating a Table of Contents Automatically" in Chapter 3 for information about this process.

```
Is your database alphabetical by heading [N] ?
```

If the headings for the articles in your database are in alphabetical order, such as in a dictionary or other reference material that is alphabetized, type Y to answer Yes to this question. If the

headings are not alphabetized, press Enter to accept the default answer of No.

The next two questions are displayed if you answer Yes to this question.

```
Use "From:" instead of range of headings [N] ?
```

Press Enter to have SEBAS generate heading ranges, for example, *alpha - axle*. Type Y if you want to display only the beginning of each range, for example, *From: Alpha.*

```
Top level menu label [Volume:] ?
```

Press Enter to use the term *volume:* to label your top level alphabetical ranges.

```
Truncate only at whole words [N] ?
```

Press Enter to truncate after 12 characters of each heading range. Type Y and press Enter to show only whole words.

```
Confirm your above choices [N] ?
```

Press Enter to repeat all of the questions, allowing you to change any or all of your previous answers. Type Y and press Enter to accept your choices and continue on to the next screen.

NOTE: The answers in the square brackets are now your previous answers and are no longer necessarily the default.

Multi Search Options

The following set of questions shows the first Multi Search screen. If at any time you answer No to the existence of a question, subsequent questions in the same table are not shown. If at any time you answer No to the existence of a table, subsequent tables are not shown, and you are asked to confirm your answers.

```
Multi search table 1 [N] ?
Multi search table 1 question 1 [N] ?
Multi search table 1 question 1 reference list [N] ?
Multi search table 1 question 1 automatic [N] ?

Multi search table 1 question 2 [N] ?
Multi search table 1 question 2 reference list [N] ?
Multi search table 1 question 2 automatic [N] ?

Multi search table 1 question 3 [N] ?
Multi search table 1 question 3 reference list [N] ?
Multi search table 1 question 3 automatic [N] ?

Multi search table 1 question 4 [N] ?
Multi search table 1 question 4 reference list [N] ?
Multi search table 1 question 4 automatic [N] ?

Multi search table 2 [N] ?
```

Type Y if you have one multi search table. Press Enter if you do not have a multi search table, or if you do not want to make a multi search index after all.

Press Enter if you do not have a question #1, and therefore no questions at all, in your first table. Type Y if you have a question #1 in your first table.

Press Enter if you do not have a reference list for question #1. Type Y if you have a reference list for your first question.

Press Enter if you are manually creating your reference list. Type Y if you want to automatically create a reference list.

Answer all of the questions in this way, until you do not have a question or a table.

These questions repeat in the exact same fashion for tables 2 through 5.

Running the BUILD.BAT Batch Program

When you have finished answering the questions in MKBATCH, it writes the BUILD.BAT batch file or other file you named onto your disk. Now you can run BUILD.BAT to create your electronic book file.

To run the build batch program, type the following command at the system prompt:

```
BUILD
```

The commands in the batch program are executed, and if you didn't redirect console output to a file, you see the messages from the build process on your screen. The amount of time it takes depends on the size of the database you are processing, and on how many indices you decided to make.

The BUILD.BAT file performs the following functions:

- Creates the graphic image block
- Creates the main text block
- Creates the indices, depending on which ones you specified
  - menu search index
  - keyword search index
  - word search index
  - endword search index
  - multi search index
  - graphic search index
- Builds the special character file
- Puts all of the files together into one file with the name you specified for your database and the extension .BLD.

You can now take your electronic book file, *database*.BLD and test it using the Data Discman Emulator program, DDMAN, described in Chapter 7.

NOTE: The MKBATCH program does not prompt you for every possible option. It might create a BUILD.BAT file that does exactly what you need, but sometimes it is necessary to edit the BUILD.BAT file and add or delete option switches or programs. Study the BUILD.BAT file and the Command Reference section of this manual to understand what the batch file is doing. Then you can edit it to suit the needs of your database.

7 Testing Your Electronic Book

This chapter tells you how to start the Data Discman Emulator program to test the electronic book you created according to the instructions in Chapters 3, 4, 5, and 6 of this user guide. You test your electronic book on a personal computer to be sure it operates correctly, before you create an XA CD-ROM formatted premaster and a CD-ROM write-once master disc.

With the exception of the exact keys to press, the emulator program works just like the electronic book player. You should be familiar with the operation of the Sony Data Discman or a functionally equivalent electronic book player, or have the manual that accompanies it. The complete operation of the Sony Data Discman electronic book player is not documented here.

Chapter 2, "Installing the Sony Electronic Book Authoring System," tells you how to use the Data Discman Emulator to work with the sample electronic books distributed with the SEBAS tools.

Starting the Data Discman Emulator

Make sure the DDMAN.EXE file is in the current path or the current directory. Also ensure that the electronic book to be accessed is in the current directory.

If you want to use a mouse while testing the electronic book, remember to install your mouse driver before you start the DDMAN program.

To start the Data Discman Emulator type the following command at the system prompt:

```
DDMAN database.BLD
```

where *database*.BLD is the name of the electronic book you created.

If you included audio segments in your electronic book, you must have a SONY PCY-311 audio encoding board to test the audio.

PCYSTART.EXE must be in the same directory as DDMAN.EXE to use the PCY-311 audio encoding board.

To start the Data Discman Emulator and test both the database and the audio file, type the following command at the system prompt:

```
DDMAN database.BLD -sSOUND
```

where *database*.BLD is the name of the electronic book you created and SOUND is the name of your sound file.

If you have created your disc directory structure on your hard disk and you are using a CATALOG file, you can change to the subdirectory that contains the CATALOG file and type the following command at the system prompt:

```
DDMAN
```

The Discman Emulator automatically finds the file CATALOG. CATALOG files are documented in Chapter 8, "Preparing to Premaster."

The Data Discman Emulator program appears on the screen, with a graphic representation of the Data Discman screen and keyboard. Use the emulator just as you would use the Data Discman. Table 7-1 shows the keys to use on the personal computer keyboard.

Test your electronic book. Make necessary changes. Test it again. When you are satisfied that everything works the way you want it to, you can move to the next step, described in Chapter 8, Preparing to Premaster.

Using the Emulator Keys

Table 7-1 shows each Data Discman key and its corresponding key on the personal computer keyboard.

Table 7-1. Data Discman Emulator Keys

Data Discman	Personal Computer Keyboard
F1, F2, F3, F4, F5	F1, F2, F3, F4, F5
NO	Esc
YES	Enter (Return on some
Left-arrow	Left-arrow or PgUp
Right-arrow	Right-arrow or PgDn
Up-arrow	Up-arrow
Down-arrow	Down-arrow
ON	Ctrl-Q
OFF	Ctrl-S or Ctrl-C

If you have a mouse, move the mouse pointer to the picture of the button on the screen, then click. The effect is the same as pressing the button.

8 Preparing to Premaster

This chapter provides a brief overview of the premastering and mastering process required to make a CD-ROM disc, and describes the steps you need to follow to set up your electronic book for premastering and mastering. Sony Corporation requires data for the Sony Data Discman electronic book player to be in the mode 1 form 1 CD-ROM XA format, referred to in this guide as the CD-ROM XA format. You can, however, test prototype databases on the Data Discman on your personal computer using an ISO 9660 formatted disc. The premastering and mastering process for ISO 9660 format is described in detail in the *ISO Formatter User's Guide* from Sony Electronic Publishing Company.

This chapter assumes you have created your electronic book file, *database*.BLD, using the instructions in Chapter 6, "Building Your Electronic Book," and that you have tested it using the instructions in Chapter 7, "Testing Your Electronic Book."

If you intend to put more than one database, that is, more than one electronic book file onto your CD-ROM master, this chapter tells you how to set up your files and directories to do that.

Premastering and Mastering

Once you have compiled a database or electronic book, you must put it into the CD-ROM XA format for premastering and replication of the disc.

Premastering, in this context, is the process of making the directory structure and data conform to the CD-ROM XA standard, and of

creating an image of the CD-ROM disc in memory or on a personal computer hard disk.

Mastering is the process of writing the image onto a CD-ROM write-once disc or a magneto optical disc. From there, the master is sent to a manufacturing or mastering facility for reproduction. Note that the manufacturing facility might refer to the CD-ROM recordable disc or magneto optical disc as their premaster.

The next section tells you the steps to follow to prepare your electronic book file or files and directory structure for premastering, and how to premaster and master using the CD-ROM XA premastering and mastering programs provided with the SEBAS tools.

NOTE: If you are using CD-ROM XA formatting software that is different from the Sony Electronic Publishing Company formatting software, or if you are using ISO formatting software, you still must use the directory structure shown in Figure 8-1. Depending on the software you use, you might have to place this structure inside some other required structure, before you make your image file. Remember to run the EBIDENT program on your premastered image file to insert the Electronic Book header used by the Discman.

Setting Up the Files and Directory Structure

To set up the files and directory structure:

1. Run the MKCATLOG program to make a CATALOG file containing the name and directory information for each database on the CD-ROM. You can have up to 7 databases on the CD-ROM. See "Making the Catalog File" later in this chapter.

2. Set up the base and subdirectory structure for formatting an electronic book image, as shown in Figure 8-1.

3. Copy the CATALOG, LANGUAGE, *database*.BLD, and interleaved SOUND file into the directory structure as shown in Figure 8-1. The LANGUAGE file is distributed with the SEBAS tools and is copied to the SEBAS\BIN directory when you install the software.

4. Rename each *database*.BLD file in each database subdirectory to START. If you have sound files, rename each sound file to SOUND.

   NOTE: Do not rename the *database*.BLD files to START and the sound file to SOUND until you have copied them into the correct subdirectory!

5. Create the CD-ROM XA image file. Refer to "Premastering and Mastering Using the Sony CD-ROM XA Tools" later in this chapter.

6. If you premastered using other CD-ROM XA premastering tools or ISO formatting tools, run the EBIDENT program to set the default menu language.

7. Use the CD-ROM XA formatter software supplied with the SEBAS tools to place the image on a CD-ROM, tape, or magneto optical disc. This is the master that you send to a manufacturing facility. Refer to "Premastering and Mastering Using Sony CD-ROM XA Tools" later in this chapter.

Making the Catalog File

You must use the MKCATLOG program to create a CATALOG file for your electronic book. The MKCATLOG program is documented in Chapter 9, "SEBAS Command Reference."

You must place the CATAOG file in the base directory for your electronic book. The base directory is the directory that contains all other directories in the electronic book.

If you have more than one database in your electronic book, the CATALOG file provides information to manage them.

Setting Up the Directory Structure

The figure below shows the proper directory structure for creating an image file for an electronic book player. The SOUND file is optional.

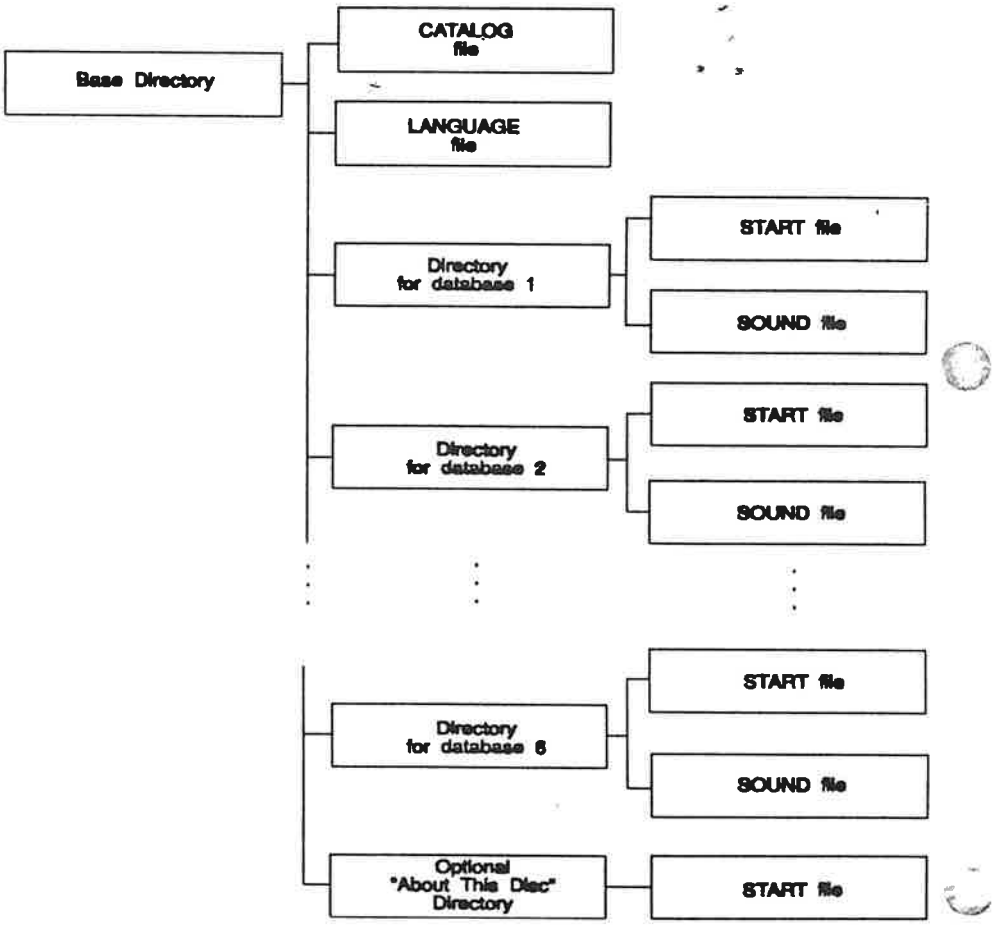

Figure 8-1. Directory Structure for Creating an Image File

Set up your directories according to the structure shown in Figure 8-1.

- The name of the base directory does not matter, but we recommend you give it a name that suggests the title of the CD-ROM volume—the title of the electronic book.
- The CATALOG and LANGUAGE files must be in the base directory.
- Directories 1 through 6 each contain one *database*.BLD file that you have renamed to START. We recommend that you give directories 1 through 7 the same name as *database* in the *database*.BLD file. These directories can also contain an optional SOUND file.
- The last directory contains the optional "About This Disc" database. It is an example of a tiny data base containing title and copyright information for the CD-ROM volume. You can make one of these for your CD-ROM. It is the only way to insert this information.
- The START files are in their respective subdirectories. Remember that each START file may have originally had the *same filename* as the directory that now contains it, with the file extension .BLD.

Testing the Directory Structure

You can use the DDMAN emulator program to test your CATALOG file and directory structure before continuing.

To use DDMAN to test the structure:

1. Change to the base directory.

2. Type the following command at the system prompt:

   DDMAN

If everything works the way you want it to, the directory structure is complete.

Once you have tested the directory structure, you are ready to use your CD-ROM XA or ISO formatting software to create a CD-ROM XA or ISO 9660 image for your electronic book on a CD-ROM.

Premastering and Mastering Using Sony CD-ROM XA Tools

The Sony CD-ROM XA software is designed to create an image file for the Sony Data Discman electronic book player. The software consists of two programs and one file: the CDISO-DM premastering program, the CDWO-DM mastering program, and the VOLUME.INF file. The CDISO-DM and CDWO-DM programs are documented in Chapter 9.

The CDISO-DM program takes a source drive or directory and creates a 2336 sector sized CD-ROM XA image file with a sub-header. The SOURCE directory must be the base directory that contains the structure of your Discman disc.

To premaster an image:

Type

```
CDISO-DM inpath outfile [infofile]
```

where

inpath is the source drive or subdirectory (for example D:\ or C:\data)

outfile is the image file to be created (such as DATA.IMG)

infofile is the name of the volume information file containing the necessary volume, XA, and Discman information. This file is needed only if the file VOLUME.INF is not in the current directory. This can be edited and the volume information can be changed for your specific application.

Example:

```
CDISO-DM C:\ABCDATA D:\ABC.IMG
```

This command creates a CD-ROM XA image file from everything in the directory C:\ABCDATA and writes it to the file C:\ABC.IMG.

The CDWO-DM program creates a write-once master disc. This program is used with the Sony CDW-W1/E1 write-once unit only.

To create a CD-ROM write-once master disc:

Type

```
CDWO-DM filename.IMG
```

where

filename is the name of the image file created by the CDISO-DM program.

Example:

```
CDWO-DM D:\ABC.IMG
```

This command creates a CD-ROM master disc from the premastered CD-ROM XA image file ABC.IMG.

Setting the Default Menu Language

The default menu language for the Sony Data Discman is set by the EBIDENT program or by premastering with the CDISO-DM program. This and other information is stored in the heading for the electronic book. You must run EBIDENT, if neccessary, *after* the premastered image file is created, and *before* it is mastered onto a Sony CD-ROM write-once disc. If you used ISO formatting software, or CD-ROM XA formatting software from a source other than Sony, then you must run the EBIDENT program to set the default menu language. You do not need to run EBIDENT if you created the CD-ROM XA image with the CDISO-DM program.

Follow these general steps to add the heading information for the menu language to your electronic book, after you make your image file:

1. Ensure that the program EBIDENT.EXE is in the current system path or the current directory, and that the image file is in the current directory.

2. Type the following command at the system prompt:

   `EBIDENT` *`filename`*`.IMG`

   where

   `filename``.IMG` is the name of the ISO 9660 or CD-ROM XA image file.

Now the premastered ISO image file is ready to be mastered onto a Sony CD-ROM write-once disc or a magneto optical cartridge. Follow the steps for mastering the image file in the *ISO Formatter User's Guide*, or follow the appropriate instructions for whatever CD-ROM XA mastering software you are using.

9 SEBAS Command Reference

This chapter describes all of the Sony Electronic Book Authoring System (SEBAS) executable commands. The MKBATCH program prompts you to create the BUILD.BAT batch file which contains the set of commands needed to process and index your book. However, you might need to use the special options available in some of these commands, or modify the commands in BUILD.BAT in some other way. Refer to Chapter 6, "Building Your Electronic Book," for information on the MKBATCH program.

The SEBAS commands are listed in alphabetical order, and contain a description of the command, the format or syntax of the command, the command options, if any, and some examples.

The input files required by each program and the output files created by each program are listed separately when they are not included in the command line syntax. Remember that input files must be in the current directory to be accessible by a command.

Table 9-1 at the end of this chapter lists the input and output files for each program. It also lists the files that can be deleted after they are used by BUILD.BAT to build your electronic book (*database*.BLD).

NOTE: When you delete files after processing, you cannot restart BUILD.BAT at an intermediate point.

Notational Conventions

The following notational conventions are used in this guide:

- `monospace font`

  You type exactly what appears in the format statement or the example as input, or the computer displays it as output.

- *`monospace font italics`*

  You must supply a filename, value, or other parameter of the type specified. An _ underbar connects words in a parameter to make it more readable while showing it is all one parameter.

- UPPER-CASE CHARACTERS

  Actual filenames are shown in all upper-case characters.

- *`database`*

  The name of your database file–the same name as your electronic book file. This is usually the same as your SGML tagged file *database*.SML, and ultimately your *database*.BLD file.

- [*`-options`*] Square brackets surround optional items on the command line. If you use an option, you must include the dash in front of the option. When options are present on the command line, each option and its description is included under the command.

- You can view the list of options for each command using the following form of the command after the system prompt:

  `COMMAND_NAME -H`

ADDSBENT

The ADDSBENT command adds the specified start block entry to your *database*.BLD file. The commands executed by the BUILD.BAT batch program create a *database*.MAP file that contains the ADDSBENT commands. This MAP file is copied to a batch file and executed at the end of BUILD.BAT.

Format:

```
ADDSBENT database.BLD id locext totblock
```

Parameters:

database	Name of the database
id	Start block entry id
locext	Location of first block
totblock	Total number of blocks

Example:

```
ADDSBENT CITYGUID.BLD msi 1025 933
```

This example shows what an ADDSBENT statement looks like in the *database*.MAP file created during the build process.

NOTE: The *locext and totblock* parameters are supplied by the build tools and this information is contained in the database.MAP file.

CDISO-DM

The CDISO-DM command creates a CD-ROM XA image file that is playable on the Sony Data Discman. Normally you do this before you create a master disc.

Format:

```
CDISO-DM inpath outfile [infofile]
```

Parameters:

inpath	The topmost source directory
outfile	Target image filename
infofile	Volume information filename

Both inpath and outfile must be specified. If infofile is not specified, the default is VOLUME.INF in the current directory.

Example:

```
CDISO-DM C:\HUCKFINN\DATA HUCKFINN.IMG
```

This command creates a CD-ROM XA image of everything in the directory HUCKFINN\DATA and its subdirectories. The resulting image file is HUCKFINN.IMG.

CDWO-DM

The CDWO-DM command writes a CD-ROM XA image file to a Sony CDW-W1/E1 write-once unit. The XA image file is normally created by the CDISO-DM program.

Format:

```
CDWO-DM [-t] [-m] filename
```

Parameters:

filename	The name of the CD-ROM XA image file.

Options:

-t	Name of file containing a valid tracklist
-m	Runs the application under the recorder's TEST mode.

Example:

```
CDWO-DM HUCKFINN.IMG
```

This command writes the file HUCKFINN.IMG to the Sony write-once unit.

CHKSUB

The CHKSUB command shows the structure of an interleaved sound file. Use it to find which sectors a channel is recorded into at a particular point in a file.

Format:

```
CHKSUB filename count [minute second frame]
```

Parameters:

filename	Interleaved sound file to check
count	Number of sectors to check (-1 for all)
minute	Beginning minute
second	Beginning second (0-59)
frame	Beginning frame (1-75)

Example

```
CHKSUB SOUND -1 10 21 35
```

This command checks all sectors and shows you the structure of the file SOUND beginning at 10:21:35. CHKSUB lists the timin (min:sec:sector), file number, channel, submode, and mode for each sector.

DDMAN

The DDMAN command invokes the Data Discman Emulator program on your personal computer, so you can test the operation of your electronic book, which is in your *database*.BLD file. DDMAN is documented in Chapter 7, Testing Your Electronic Book.

Format:

```
DDMAN [database.BLD] [-options]
```

Parameters:

database.BLD	Name of your database file—your electronic book file, after it has been built by the BUILD.BAT batch program.
no parameter	DDMAN looks first for a CATALOG file, then for a START file in the same directory.

Options:

-s *filename*	Name of the sound file you are using

Examples:

```
DDMAN DICTNARY.BLD
```

This command invokes the Data Discman Emulator with the electronic book file DICTNARY.

```
DDMAN DICTNARY.BLD -sSOUND
```

This command invokes the Data Discman Emulator with the electronic book file DICTNARY and the associated sound file SOUND.

DDMAN

Use this form of the command to test the operation of your CATALOG file and directory structure, before creating your image file.

DMPPCX

The DMPPCX command displays a PCX graphic on your personal computer as it will appear on the electronic book player, in electronic book image (EB) format.

Format:

```
DMPPCX filename.PCX
```

Parameters:

filename.PCX Name of your PCX graphic file

Example:

```
DMPPCX BLANK.PCX
```

This command displays the following screen of information about the graphic BLANK.PCX. There is a menu at the bottom from which you can pick a function.

```
**** SONY PCX Dump Utility:  Version 1.40 ****
Copyright (c)  1991 SONY Corporation  All Righ
Reserved.
PCX Header:
Manufacturer              =     10
Version                   =      3
Run Length Encoding       =      1
Bits Per Pixel            =      1
Upper Left Corner:  X     =      0
                    Y     =      0
Lower Right Corner: X     =    255
                    Y     =    199
Display: Horizontal Res   =    640
         Vertical   Res   =    480
Number of Planes          =      1
Bytes Per Line            =     32
Palette Information       =   2992
DitherBuckets[5]          =   0   0   0   0   0
DitherThresholds[5]       =   0   0   0   0   0
```

```
Command: D)ump, H)eader, O)ptimal-Toggle,
P)arams, Q)uit, R)andom Dither, S)how,
T)oggle-Dither,              [H]?
```

The following list shows the meaning of the menu commands.

D	Dumps the hexadecimal representation of the image
H	Displays the PCX header
O	Toggles the optional threshold on and off
P	Normally sets the threshold and RGB multipliers
Q	Quits the DMPPCX program
R	Toggles random dithering on and off
S	Shows the PCX graphic as it would look in electronic book format on an electronic book player
T	Toggles dithering on and off

DMPVISTA

The DMPVISTA command displays a VISTA file on your personal computer in electronic book image (EBI) format.

Format:

```
DMPVISTA filename.TGA
```

Parameters:

filename.TGA	Name of your VISTA graphic file

Example:

```
DMPVISTA TRUMPET.TGA
```

This command displays the TGA graphic TRUMPET as it will appear on an electronic book player.

EB2PCX

The EB2PCX command converts an electronic book (EBI) image to a PCX file for editing.

Format:

```
EB2PCX filename.EB filename.PCX [-options]
```

Parameters:

filename.EB	Name of your electronic book graphic file that you want to convert
filename.PCX	Name of your PCX graphic file that you are converting to

Example:

```
EB2PCX TRUMPET.EB TRUMPET.PCX
```

This command converts the graphic TRUMPET.EB from electronic book image format to PCX graphic format for editing on a personal computer.

EBIDENT

The EBIDENT command sets the default menu language for the electronic book. You need to run EBIDENT only if you did not use the Sony CD-ROM XA formatting program (CDISO-DM) to premaster your image file. You must run the EBIDENT command *after* you make a premaster CD-ROM XA image file, and *before* you write the image onto a master.

Detailed information about this process appears in Chapter 8, Preparing to Premaster. The *ISO Formatter User's Guide* from Sony Electronic Publishing Company describes the premastering and mastering process for the ISO 9660 format in detail.

Format:

```
EBIDENT database.IMG [-options]
```

Parameters:

`database.IMG`	Name of the formatted image file

Options:

SSSS	Optional four-digit sector size of image file; default is 2048 bytes.
N	Optional number of language choice; default is English.
0	Japanese
1	English
2	French
3	German
4	Italian
5	Spanish
6-9	reserved

Example:

```
EBIDENT WELLNESS.IMG
```

This command sets the menu language to English.

EXTWORD

The EXTWORD command extracts keywords from each article.

Format:

```
EXTWORD database [-options]
```

Parameters:

database	Name of your database file

Options:

-A	Includes all heading words in keyword index (KWIC Permutation, use with the -A switch on WSHEAD)
-B	Removes non-alphanumerics from heading start/end (instead of just spaces from the front of the heading and spaces and hyphens from the back)
-C	Suppresses create start block
-F*n*	Fence for indexing *n* = article number (default = 1) (skip TOC)
-H	Help on options (this screen)
-I	Does not include headings in keyword index (default is to include headings)
K	Indexes headings only for keywords
M*nq*	Multi search question index extraction *n* = table number *q* = question number
O*n*	Options for hyphen processing; *n* = 0, 1, or 2: 0 (Default) treats hyphen as alphabetic, requires hyphen entry to match word 1 Treats hyphen as white space

2	Hyphen is both alphabetic and white space, before hyphenated and each word
P	Disables the percent of progress message in batch mode
W	Suppresses word extraction (no EXT), build HDG only
X	Excludes numbers from keyword index
1	Sets minimum word length set to 1 (default is 2)
Z	Includes the document text in the word and endword search index

Example:

```
EXTWORD DICTNARY
```

This command extracts keywords and headings from each article in the database file DICTNARY.

NOTE: STOPWORD file is used if present; otherwise the KEYWORD file is used if present.

FHFIXUP

Targets of text links must be headings to comply with the Sony Electronic Book specification. The FHFIXUP command is used to convert fake headings" inserted by STF2EB (using the -H2 switch) back into real headings.

Only <h1> headings are true headings in the electronic book, so any book created using MKEMENUS/MRGMENUS, or for some other reason has text targets other than <h1> headings does not meet the Sony Technical Specification for electronic books.

There are several ways to correct this:

- Change all targets of text links to <h1> headings.
- Change the definitions of <h2>, <h3>, ... headings to be the same as <h1> headings in the PUBLIC.DTD file. This may have undesirable side effects, as the headings appear in the match lists of word, keyword and multi searches.
- Use the -H2 switch in STF2EB, which makes STF2EB treat non-<h1> headings as if they are <h1> headings, but puts "fake headings" into the file it creates, so that other SEBAS tools won't recognize them as headings when building indices. If you use this switch, you must run FHFIXUP at the end of the build process to convert the fake headings back into real ones.

Format:

```
FHFIXUP filename.BLD
```

Parameters:

filename.BLD	Name of your electronic book file

Example:

```
FHFIXUP DICTNARY.BLD
```

This command converts the fake headings in DICTNARY.BLD to real headings according to the Sony Technical Specification.

FIF2EB

The FIF2EB command uses the figure input file and builds an electronic book with a graphic search. The figure input file contains the name of each graphic electronic book image, and is described in Appendix A, Creating an Electronic Slideshow.

Format:

```
FIF2EB database.FIF database.EBI [-options]
```

Parameters:

database.FIF	Name of the figure input file. The .FIF file extension is required.
database.EBI	Name of the electronic book image file. The file extension .EBI is not required, but is recommended for easy recognition.

Options:

-G	Graphics search uses actual links (IXY file) rather than having FIF2EB create sequential links.
-H	Help on options (this screen)
-I	Enables IXY file. IXY files are described in Chapter 4, "Preparing Your Graphics."
-P	Disables the percent of progress message in batch mode
-S	Suppresses graphics search in main menu

Example:

```
FIF2EB FIGURES.FIF FIGURES.EBI -I
```

This command processes the FIGURES.FIF file and enables the IXY file option.

FONTS2EB

The FONTS2EB command creates a font block containing your special characters. Special characters are described in Appendix C, Creating and Using Special Characters.

Format:

```
FONTS2EB database [-options]
```

Parameters:

database	Name of the database

Options:

-H	Help on options (this screen)
-P	Disables percent of progress message in batch mode
-Sn	Start block number; n is the number of the start block

Example:

```
FONTS2EB DICTNARY
```

HDG2HSR

The HDG2HSR command produces an HSR file from an HDG file that is in alphabetical order.

Format:

```
HDG2HSR database [-options]
```

Parameters:

database	Name of the database file

Options:

-H	Help on options (this screen)
-P	Disables the percent of progress message in batch mode
-W	Suppresses "skipping" messages. Skipping messages occurs when the same word is found more than 1500 times while creating an index (750 times for multi search index). The SEBAS tools normally skip such words and display a warning message to this effect.

Example:

```
HDG2HSR DICTNARY
```

This example produces the file DICNARY.HSR

HSR2MHD

The HSR2MHD command builds the MHD menu search heading file from an HSR sorted heading file. HSR2MHD allows for flexible generation of menus. HSR2MHD allows for numeric as well as alphabetic headings, and lets you include headings beginning with accented characters. You can also specify the appearance of the top level of the menu, in case you don't like the default (A,B,C,D and E,F,G,H and so forth).

Headings that begin with a number are treated as numeric values. Alternatively, you can specify where in the heading the number begins, and how long it is.

Information on what the top level menu looks like and how it is organized is stored in the control file HSR2MHD.MNU. In the control file, one line corresponds to each line of the menu. In the first 39 columns, the text of the menu item appears, just as it will appear on the screen. If you want the first and/or last heading to be used, rather than indicating the exact text of the menu item, you can use %F and %L, respectively. For example:

Sample Control File HSR2MHD.MNU

```
0 - 9999                0 - 9999
```

will display the menu item "0 - 9999." The range of numbers that will fit into that menu choice is within 0 to 9999.

Assume that the first heading greater than 0 is 101, and the last heading less than 9999 is 6823.

Then:

```
%F - %L                 0 - 9999
```

will display the menu item "101 - 6823". The range of numbers that fits into that menu choice is 0 to 9999.

You can use an asterisk to represent an infinitely large number, as in "0-*", which includes all numbers in the range.

Column 40 begins a field that indicates which headings will be grouped together under this menu heading. This can consist of a letter (A), a list of letters (A,B,C,D), a range of letters (A-F), or a range of numbers (0-9999).

Note that accented characters can be included in the field, but if they are, they must agree with the way the heading file was sorted. For example, if the A and Ä (umlaut) characters are sorted so that the A headings precede the Ä headings, the following line can appear in a control file:

Term: A,Ä,B,C	A, Ä, B, C

Sample Control File

Options On Multi Search Screen	Ranges That Fall Into These Options
0 - 9999	0-9999
10000 - %L	10000-*
Term: A,B,C	A-C
Term: D,E,F	D-F
Term: G,H,I,J,K	G-K
Term: L,M,N	L-N
Term: O,P,Q,R	O-R
Term: S,T	S-T
Term: U,V,W,X,Y,Z	U-Z

Format:

```
HSR2MHD database.HSR [-options]
```

Parameters:

database.HSR	Name of the database file with .HSR extension

Options:

-A	Alternates string to use with Bucket suppress. Maximum length 20 characters; default is "From:"
-B	Bucket suppress, use From:
-C*fn*	Reads control information from the file *fn*
-E	Breaks up the headings into evenly spaced groups. This option can be used if the headings don't appear in an easily detectable order, such as part numbers.
-F	Forces menu search (A-Z)
-H	Help on options (this screen)
-Gn	When extracting a numeric value from a heading, use n digits. Without this switch, the numeric value consists of all sequential digits up to the first non-numeric character.
-Kn	Skips n digits from beginning of heading before extracting numeric value.
-KCc	Skips through character c in heading before extracting numeric value.
-KF	Skips until first numeric character in heading before extracting numeric value.
-Ln	Puts at most n lines in each menu created. Default is 8, to fit on the screen without scrolling at the top level, the number of lines in the menu is determined by the number of lines in the control file.

`-M "string"`
: Menu label for top level string; can be up to 12 chars surrounded by double quotes. Defaults to "Volume:"

`-P`
: Disables the percent of progress message in batch mode

`-R`
: Generates reference table. Making reference tables is documented in Chapter 3.

`-T`
: Truncates after whole words only

`-W`
: Disables warning messages

Example:

```
HSR2MHD DICTNARY -R -Cdict.mnu
```

This command creates a DICTNARY.MHD file to be further indexed into a reference table. It has a specified control file dict.mnu to be used.

INTERLV

The INTERLV command interleaves up to 16 PCY-311 audio files specified in the listfile. The audio files can be a mixture of B stereo, B mono, C stereo, and C mono. Refer to "Preparing Your Audio" in Chapter 5 for more information.

Format:

```
INTERLV listfile output-filename
```

Parameters:

listfile Name of the text file containing the names of the audio files to be interleaved.

output-filename Name you give the resulting interleaved file

Example:

```
INTERLV AUDIO.LST SEGMENT1.SND
```

This command interleaves up to 16 PCY-311 files listed in AUDIO.LST, creating a file called SEGMENT1.SND and a cue file, SEGMENT1.CUE containing the beginning and ending timings for each input file.

KSHEAD

The KSHEAD command builds a heading block for keyword search.

Format:

```
KSHEAD database [-options]
```

Parameters:

database	Name of the database file

Options:

-Fn	Maximum frequency of occurrences (n) of a word before it is skipped and not indexed (default is 1500 for all searches except multi search, where default is 750 occurrences.
-H	Help on options (this screen)
-M	Does not generate a .KSH file in Multi search
-P	Disables the percent of progress message in batch mode
-W	Suppresses warnings for "skipping" messages

Example:

```
KSHEAD CITYGUID
```

This command makes an index for keywords in the database called CITYGUID.

KSINDEX

The KSINDEX command builds a keyword search index file.

Format:

```
KSINDEX database [-options]
```

Parameters:

database	Name of the database file

Options:

-H	Help on options (this screen)
-M*nq*	Multi Search index creation n = number q = question
-P	Disables percent of progress message in batch mode

Example:

```
KSINDEX TIMETABL
```

Input Files:

database.WHD

database.MAP

Output Files:

database.KSI

database.MAP

MHD2MSI

The MHD2MSI command creates an MSI menu search index file from an MHD menu headings file.

Format:

```
MHD2MSI database [-options]
```

Parameters:

database	Name of the database file

Options:

-H	Help on options (this screen)
-P	Disables percent of process message in batch mode

Example:

```
MHD2MSI WORLDFCT
```

This command creates the menu search index file WORLDFCT.MSI.

MHD2RMQ

The MHD2RMQ command creates a multi search index file from a MHD menu headings file.

Format:

```
MHD2RMQ database [-options]
```

Parameters:

database	Name of the database file

Options:

-H	Help on options (this screen)
-M*nq*	Multi Search reference table n = Multi Search table number q = question number
-P	Disables percent progress messages in batch mode
-S*n*	Start block *n*

Example:

```
MHD2RMQ CAREERS -M11
```

This command creates the reference table for multi search table 1 question 1.

MKBATCH

The MKBATCH command displays a series of questions that you answer. MKBATCH constructs the BUILD.BAT file according to your answers. The BUILD.BAT file contains the commands to process your tagged SGML file, generate your requested indices, and build all of the files into one electronic book file called *database*.BLD. The MKBATCH command is documented in detail in Chapter 6, Building Your Electronic Book.

Format:

```
MKBATCH
```

Example:

```
MKBATCH
```

This command invokes the MKBATCH command and displays the first set of questions. Press the Enter key to accept all of the defaults. Type Y to confirm answers.

MKCATLOG

The MKCATLOG command makes an electronic book catalog file that contains the name of each database on the CD-ROM. The catalog file must be placed in the base directory before you create an image of your book. See Chapter 8, Preparing to Premaster, for more information.

Run MKCATLOG once for each database to be included on the CD-ROM. MKCATLOG appends the new information to the catalog file each time.

Format:

```
MKCATLOG filename "database_title" directory
```

Parameters:

filename	Name of the catalog file, usually CATALOG for electronic books
database_title	Name that appears on the opening menu of the electronic book player, showing the name of the database. It must be less than or equal to 30 characters.
directory	Name of the subdirectory containing the START file for the database.

Example:

```
MKCATLOG CATALOG "About This Disc" INFO
```

This command makes the catalog file named CATALOG. The database title is "About This Disc." The subdirectory name is INFO.

MKEMENUS

The MKEMENUS command builds a hierarchical browser using consultation links. It is a pre-processor to the MRGMENUS and SGML2STF programs. It adds a menu structure to the top of your .SML file before you run SGML2STF.

Format:

```
MKEMENUS database.SML TMP.NIT TMP.NTR [-options]
```

Parameters:

file.SML	Filename of file to process
TMP.NIT	Output filename
TMP.NTR	Note reference output filename

Options:

-H	Help on options (this screen)
-P	Disables percent of progress message in batch mode
-B*n*	Sets beginning line number; default=0
-E*n*	Sets ending line number; default=EOF
-Y*n*	Sets beginning level; default=0
-Z*n*	Sets ending level; default=11
-L*name*	Labels at the front of each nit and ntr; default = none

Example:

```
MKEMENUS DICTNARY.SML TMP.NIT TMP.NTR
```

This command creates the source and target menu files from the DICTNARY.SML file.

MKMAP

The MKMAP command creates a MAP file with a single entity for the main text block (the whole database). EXTWORD normally creates the database.MAP file. If you do not use EXTWORD and you still need a database.MAP file use MKMAP to create it.

Format:

```
MKMAP database [-options]
```

Parameters:

database	Name of the database file

Options:

-H	Help on options (this screen)
-N*n*	Number of blocks *n*
-S*n*	Starts block *n*

Example:

```
MKMAP DICTNARY
```

This command makes a DICTNARY.MAP file.

MKSOUND

The MKSOUND program concatenates the interleaved files specified in the listfile. These files must have been created with the INTERLV or SINTERLV program.

Format:

```
MKSOUND listfile output-filename
```

Parameters:

listfile Name of the text file containing the names of the interleaved files to be concatenated

output-filename Name you give the resulting file

Example:

```
MKSOUND AUDIO.LST SOUND
```

This command concatenates the interleaved files listed in AUDIO.LST, creating a file called SOUND and a master cue file called SOUND.CUE that lists the start and end points for each file in AUDIO.LST.

Note that the output file does not have to be called SOUND at this point, but it must be renamed the SOUND before your final image is built, when you copy it into the subdirectory with your START file.

MRGMENUS

The MRGMENUS command merges hierarchical browsing menus. You must run MKEMENUS first, because MRGMENUS uses the output from MKEMENUS.

Format:

```
MRGMENUS TMP.NIT TMP.NTS TMP.SMX *[-options]
```

Parameters:

`TMP.SML`	SGML file to process
`TMP.NTS`	Input file containing note-in-text-reference-id tags <ntr rid=*label-for-source*> `TMP.NTS` is the sorted `TMP.NTR` file
`TMP.SMX`	Output filename

Options:

`-H`	Help on options (this screen)
`-P`	Disables percent of progress in batch mode
`-T`	Does NOT output table of contents. Use this option when you are going to concatenate several files together and therefore do not want multiple <h1> Table of Contents.
`-X`	Does NOT do ntr word wrap
`-I`*n*	*n* = number of spaces to indent second lines of menu strings (default=2)
`-C`*n*	*n* = number of columns for a menu (default=27)

Example:

```
MRGMENUS TMP.NIT TMP.NTS DICTNARY.SMX
```

This command uses the NIT output and the sorted NTS output from the MKEMENUS command, and creates a new input for SGML2STF, called DICTNARY.SMX, from which SGML2STF builds a database file.

MSFIXUP

The MSFIXUP command fixes all multi search tags in main text by writing over the multi search tags with spaces.

Format:

```
MSFIXUP database.BLD
```

Parameters:

database.BLD	Name of the output database file after running the BUILD.BAT program

Example:

```
MSFIXUP CITYGUID.BLD
```

This example eliminates all the {MS} tags from CITYGUID.BLD.

MSREFTAB

The MSREFTAB command builds the multi search reference tables.

Format:

```
MSREFTAB database -Mnq [-options]
```

Parameters:

database	Name of the database file
-M*nq*	Multi Search reference table *n* = table number *q* = question number

Options:

-H	Help on options (this screen)
-P	Disables percent of progress messages in batch mode
-S*n*	Starts block *n*

Example:

```
MSREFTAB CITYGUID -M54
```

This command builds a multi search reference table for table #5 and question #4 of the database named CITYGUID.

MSTABLE

The MSTABLE command builds the multi search table of questions.

Format:

```
MSREFTAB database -Mn [-options]
```

Parameters:

database	Name of the database file
-M*n*	Multi search reference table *n* = table number

Options:

-H	Help on options (this screen)
-P	Disables percent of progress messages in batch mode
-S*n*	Starts block *n*

Example:

```
MSTABLE CITYGUID -M5
```

This example builds multi search table #5 for CITYGUID.

NULLFILE

The NULLFILE command creates a null length file.

Format:

```
NULLFILE database
```

Parameters:

database	Name of the database file

Example:

```
NULLFILE DICTNARY
```

This example creates a null length file called DICTNARY.

OPTSORT

The OPTSORT command sorts files using the Opt-Tech sort program licensed by Sony. You can use a different sort program that performs the same function. The primary sort done by the SEBAS batch job is: alphanumeric, ascending, variable length record, and conversion to upper case.

Format:

```
OPTSORT input_file output_file control_file
```

Parameters:

input_file	Input file
output_file	Output file
control_file	OPTSORT.CTL, HLTSORT.CTL, or MHD2RMQ.CTL

Example:

```
OPTSORT CITYGUID.HDG CITYGUID.HSR OPTSORT.CTL
```

This example sorts the CITYGUID.HDG file into CITYGUID.HSR using the OPTSORT.CTL control file.

PCX2EB

The PCX2EB command converts a PCX file to electronic book image (EB) file format.

Format:

```
PCX2EB filename.PCX filename.EBI [-options]
```

Parameters:

filename.PCX	Name of PCX file to convert
filename.EBI	Name of EBI file to convert to

Options:

-B	Blue factor
-C	Centering off
-G	Green Factor
-H	Help on options (this screen)
-O	Optimal threshold (no dither)
-M	Random dithering
-P	Disables percent of progress messages in batch mode
-R	Red factor
-S	Stippling (Dither)
-T	Threshold
-2	2X small images

Example:

```
PCX2EB TRUMPET.PCX TRUMPET.EB
```

This example converts the figure TRUMPET from PCX to EB format.

PCYEDIT

The PCYEDIT command allows you to manually trim the beginning and/or end of an audio segment by sectors.

On-Line Keys

F1	Views help screen
F3	Goes to previous file
F4	Goes to next file
F5	Removes a sector from the beginning
F6	Replaces a sector to the beginning
F7	Removes a sector from the end
F8	Replaces a sector to the end
F9	Saves current changes
F10 or ESC	Exits
+	Lists previously played audio file in log file
ENTER	Plays audio file

NOTE: While playing press any key to stop. Pressing F3 or F4 automatically saves changes to a file.

Format:

```
PCYEDIT filename.PCY [filenam2.PCY
                     filenam3.PCY...]
```

or

```
PCYEDIT @listfile
```

Parameters:

filename.PCY	The raw non-interleaved sound file to play. Wild cards are accepted.
listfile	A plain text file containing the list of PCY sound files to edit

Example:

```
PCYEDIT CHAPTER2.PCY
```

This command trims a PCY-311 sound file named CHAPTER2.PCY.

F5 trims a sector at a time from the beginning of the sound file, and F7 trims a sector at a time from the end of the file. F6 and F8 can be used to restore sectors deleted with F5 and F7 during the same editing session. You can play the trimmed audio file by pressing Enter.

The trimmed file is saved with the .TRM extension and the original file is left untouched (unless, of course, your original file has the .TRM extension).

PCYHDR

The PCYHDR command reports the header information in a sound file. It gives the version, offset, size in bytes, block size, mode, file format, compression, rate in hertz, bitwidth, channels, and emphasis.

Format:

```
PCYHDR filename.PCY
```

Parameters:

filename.PCY Name of a PCY-311 sound file

Example:

```
PCYHDR CHAPTER3.PCY
```

This command reports the header information from the sound file CHAPTER3.PCY onto the screen.

PCYLEN

The PCYLEN command tells you the length in minutes, seconds, and frames of a PCY-311 sound file.

Format:

```
PCYLEN filename.PCY
```

or

```
@listfile
```

Parameters:

filename.PCY	Name of a PCY-311 sound file
listfile	File containing a list of PCY-311 sound files

Example:

```
PCYLEN CHAPTER3.PCY
```

This command reports the length of the sound file CHAPTER3.PCY.

PCYPLAY

The PCYPLAY command plays a PCY-311 sound file.

Format:

```
PCYPLAY filename.PCY [filenam2.PCY
                     filenam3.PCY...]
```

or

```
PCYPLAY @listfile
```

Parameters:

filename.PCY	The PCY sound file to play. Wild cards are accepted.
listfile	A plain text file containing the list of PCY sound files to play

Example:

```
PCYPLAY AUDSEG1.PCY
```

This command displays a screen that allows you to play the PCY-311 sound file AUDSEG1.PCY. This program is similar to the DECODE program that comes with your PCY-311 board, except that DECODE does not accept wild cards or list files.

PCYREC

The PCYREC command records sound files through the PCY-311 board in all XA modes.

Format:

```
PCYREC filename.PCY [mode]
```

Parameters:

filename.PCY	PCY sound file to record

Option:

mode	The XA mode to record in. Can be any of the following:
cm	C mono=(default)
cme	C mono, emphasis on
cs	C stereo
cse	C stereo, emphasis on
bm	B mono
bme	B mono, emphasis on
bs	B stereo
bse	B stereo, emphasis on
am	A mono
ame	A mono, emphasis on
as	A stereo
ase	A stereo, emphasis on

Example:

```
PCYREC AUDSEG2.PCY -bm
```

This command allows you to record a sound file named AUDSEG2.PCY in the B mono mode.

PCYSPLIT

The PCYSPLIT command splits a CD-ROM XA formatted sound file into two or more output files according to the input time indices.

Format:

```
PCYSPLIT InputFile
```

Parameters:

InputFile — The file containing the list of parameters for PCYSPLIT.

The `InputFile` contents are as follows:

line 1: input PCY filename

line 2: OutputFileName mm:ss:ff

.

.

.

line n: outputFileName mm:ss:ff

where mm:ss:ff are the ending minute, second, and frame.

Example:

```
PCYSPLIT SOUNDLST
```

This command splits a CD-ROM XA formatted sound file according to the parameters in the inputfile SOUNDLST.

PLAYER

The PLAYER command plays a sequenced list of sound files, normally created with the RECORDER program. The PLAYER command also allows you to log the previously played audio file into a log file for future reference.

On-Line Keys

F1	Views help screen
F3	Goes to previous file
F4	Goes to next file
F10 or ESC	Exits
+	Lists previously played audio file in log file
ENTER	Plays audio file and simultaneously advance to the next file

NOTE: While playing press any key to stop.

Format:

```
PLAYER filename [logfile]
```

Parameters:

filename	The first file to be played. The name should be in the format bbbbnnnn where bbbb is the base name and nnnn is the starting file number.
logfile	Contains a log of errors entered by the user. Default name is PLAYER.LOG.

Example:

```
PLAYER HUCK0001
```

This command displays a screen that allows you to play a sequence of files beginning with HUCK0001.

Use the + key on the numeric keypad to enter error comments.

PLAYXA

The PLAYXA command plays user-defined sections of an interleaved CD-ROM XA formatted sound file.

On-Line Keys

F1	Views help screen
F2	Toggles EOR (end of recording) on or off
F3	Goes down one channel
F4	Goes up one channel
F5	Subtracts frames from start position
F6	Adds frames to start position
F7	Subtracts seconds from start position
F8	Adds seconds to start position
F10 or ESC	Exits
ENTER	Plays audio file

NOTE: While playing press any key to stop.. If you want to hear more than the first section you must turn off the EOR flag.

Format:

```
PLAYXA filename
```

Parameters:

filename	Interleaved CD-ROM XA formatted sound file to play

Example:

PLAYXA JONESAUD

This command displays a screen that allows you to set the channel and start and stop times of an audio section you want to hear, then allows you to play back that section through the PCY-

311 audio board. A Sony PCY-311 audio board is required for you to use this program.

RECORDER

The RECORDER command records a sequence of sound files.

On-Line Keys

F1	Views help screen
F2	Changes recorder mode
F3	Goes down one file
F4	Goes up one file
F5	Lowers left record volume
F6	Raises left record volume
F7	Lowers right record volume
F8	Raises right record volume
F10 or ESC	Exits
ENTER	Starts record

NOTE: While recording press any key to stop.

Format:

```
RECORDER filename
```

Parameters:

filename — The name of the first file to record. This name must be in the format:

bbbbnnnn

where bbbb is the base name and nnnn is the starting file number.

Example:

```
RECORDER HUCK0001
```

This command displays a screen that allows you to record a series of PCY-311 files. The first file is be named HUCK0001.ADP; the second is HUCK0002.ADP; and so on. These filenames are the default: you can increment or decrement the number from within the program by using the F3 or F4 key. You can also adjust the recording volume from the program screen.

RERECORD

The RERECORD command rerecords a group of sound files.

On-Line Keys

F1	Views help screen
F2	Changes recorder mode
F3	Goes down one file
F4	Goes up one file
F5	Lowers left record volume
F6	Raises left record volume
F7	Lowers right record volume
F8	Raises right record volume
F10 or ESC	Exits
ENTER	Starts record

NOTE: While recording press any key to stop.

Format:

```
RERECORD LogFileName
```

Parameters:

LogFileName	Log file created by the PLAYER program

Example:

```
RERECORD PLAYER.LOG
```

This command displays a screen that allows you to rerecord the files in the log file PLAYER.LOG.

REVERSE

The REVERSE command reverses headings in order to create an index for endword search.

Format:

```
REVERSE database.HHD [-options]
```

Parameters:

database.HHD	File produced by the EXTWORD command

Options:

-A	Includes all words in headings in the endword search (KWIC permutation)
-H	Help on options (this screen)

Example:

```
REVERSE CITYGUID.HHD
```

This example reverses all the headings in the CITYGUID.HHD file.

RPLSBENT

The RPLSBENT command replaces an existing start block entry with the new specified values.

It is possible to create a new menu structure after the build process. If a new menu structure is created it can be copied to the end of database.BLD file. Use RPLSBENT to update the pointers to the new menu structure.

Format:

```
RPLSBENT filename id locext totblock
```

Parameters:

filename	Name of the database
id	Start block entry id
locext	Location of first block
totblock	Total number of blocks

Example:

```
RPLSBENT msi 1304 965
```

This command updates the start block to include the new menu search that has been added.

SGML2STF

The SGML2STF command converts the tagged Standard Generalized Markup Language (SGML) file (*database*.SML) to the SGML Token File format (*database*.STF), for further processing by the SEBAS tools. SGML2STF checks that all of the SGML tags are properly formatted.

This is the first SEBAS tool you use to process your electronic book after you tag the original files. If the SGML2STF program shows that your file contains syntax errors in tagging, you must correct the errors and run SGML2STF again, until your file is without syntax errors.

SGML2STF will be run again as one of the first steps in the BUILD.BAT file. At that point you will want it to run without errors or warnings.

Format:

```
SGML2STF database.SML [-options]
```

Parameters:

database.SML	SGML tagged file

Options:

C	Does not Create an STF file
E	Does not write STF output after Errors
L	Creates listing to file *database*.LST
O	Help (this screen)
Q	Quiet mode; supresses screen display of statistical information, displays warnings and errors
S	STF output to STDOUT
	Overrides l option
U	Update mode (Do Not Use)

v	Turns off setvbuf for files
w	Writes to a Window
x	Enables user exit

Example:

```
SGML2STF COOKBOOK.SML -q -w
```

This command parses the COOKBOOK.SML file into COOKBOOK.STF. It supresses statistical information, and writes the output to a window.

SHOWEBI

The SHOWEBI command displays multiple electronic book image files. You can specify up to four filenames on the command line.

Format:

```
SHOWEBI filename.EB [,filename.EB, ...]
```

Parameters:

filename.EB Name of the electronic book image file

Example:

```
SHOWEBI TRUMPET.EB
```

This command displays the graphic TRUMPET.EB on the screen.

SINTERLV

The SINTERLV command interleaves the PCY-311 audio files specified in a list file and places them in a named output file. All audio files must be in C mono.

SINTERLV creates a sound file, a cue file with the extension .CUE, and a link information file for each channel. The link information file has the same name as the input filename, with the extension .XA. This program optimizes space for the audio files.

Format:

```
SINTERLV listfile output-filename
```

Parameters:

listfile — Name of the text file containing the names of the audio files to be interleaved.

output-filename — Name of the resulting interleaved file

Example:

```
SINTERLV AUDIO.LST MASTER
```

This command interleaves all the PCY-311 files listed in AUDIO.LST, and creates a file called MASTER, a cue file called MASTER.CUE, and a link information file with the extension .XA for each file in AUDIO.LST.

Note that the output file must be renamed to SOUND before your final image is built, when you copy it into the subdirectory with your START file.

STF2EB

The STF2EB command converts an SGML token file to the electronic book text block format. STF2EB uses all available memory for consultation link targets. You can process approximately 8000 links without using expanded memory (EMS). With an EMS driver loaded, you can process approximately 300,000 links for every 8 megabytes of EMS.

Format:

```
STF2EB database.STF database.EBI [-options]
```

Parameters:

database.STF	SGML token file
database.EBI	Electronic book text block format

Options:

-A	Supresses article separator (row of dashes)
-B1	Enables browse from article to next article
-F*fn*	Figure list, *fn* is filename
-G	Graphic links (IXY files) present
-H	Help on options (this screen)
-H2	Creates "fake" headings for h2 and lower heading levels. (Requires running FHFIXUP on database.BLD file to restore real headings.)
-I*fn*	Image start block, *fn* is filename
-J*n*	Sets literal indent to *n* n=0 or 1; default is 0
-K	Supresses linefeed after figure links
-L1	Inserts linefeed after text links

`-Mn`	Margin of extra hard spaces for first line of paragraph; default is 0
`-N1`	Enables Next / Previous article links
`-O1`	Enables outline link from article
`-P`	Disables percent of progress messages in batch mode
`-QAc`	Defines c to be article separator; default is "-"
`-QHn`	Indents heading n number of spaces
`-S`*fn*	Sorted HLT's; *fn* is filename
`-T`	Table of contents; (1st article)
`-U`	Forces blank line after article headings
`-W`	Disables warnings

Example:

```
STF2EB IDIOMS.STF IDIOMS
```

This command converts the SGML token file IDIOMS.STF to the electronic book file IDIOMS.EBI.

VISTA2EB

The VISTA2EB command converts a VISTA file to a file in electronic book image (EB) format.

Format:

```
VISTA2EB filename.TGA filename.EB [-options]
```

Parameters:

filename.TGA	Name of Vista (Targa) file to convert from
filename.EB	Name of electronic book image file to convert to

Options:

-B	Blue Factor
-G	Green Factor
-H	Help on options (this screen)
-O	Optimal threshold (no dither)
-M	Random dithering
-P	Disables percent of progress messages for batch mode
-R	Red Factor
-S	Stippling (Dither)
-T	Threshold

Example:

```
VISTA2EB VIOLIN.TGA VIOLIN.EB
```

This command converts the file VIOLIN.TGA to the file VIOLIN.EB.

WSHEAD

The WSHEAD command builds a headings file for word and endword searches.

Format:

```
WSHEAD database [-options]
```

Parameters:

database	Name of database file

Options:

-A	Includes all heading words in word search (KWIC permutation)
-C	Does not convert foreign characters not on keyboard
-D	Debug mode
-E	Makes endword search headings entry
-H	Help on options (this screen)
-P	Disables percent of progress messages for batch mode
-R	Does not remove punctuation
-S*n*	Start block *n*
-Z	Converts special characters to ISO equivalents

Example:

```
WSHEAD IDIOMS
```

This example builds the headings file for word and endword searches for the database named IDIOMS.

WSINDEX

The WSINDEX command builds an index file for word and endword search.

Format:

```
WSINDEX database mode [-Sn]
```

Parameters:

database	Name of the database file
WS	Mode for word search index
ES	Mode for endword search index

Options:

Sn	Start block number n.

Example:

```
WSINDEX IDIOMS
```

This example builds an index file for word and endword search for the database named IDIOMS.

Input and Output Files

Table 9-1 shows the input, output, and deletable files for the commands in the SEBAS tool set. The input files need to be accessible on disk for the program to work properly. Output files are created by the program.

Files you can delete are files that can be deleted after they are used by the BUILD.BAT batch process. The only reason for doing this is to use the minimal amount of disk space necessary to create the electronic book. However, this is NOT recommended because it deletes all intermediate files so that you cannot restart the build at an intermediate point.

Table 9-1. Input and Output Files

Program	Input	Output	Files you can delete after build
SGML2STF	PUBLIC.DTD		Note: * marks files for database.BLD
	database.DTD		
	database.REF		
	database.SML		Can be deleted
	database.SYS		
		database.FIF	Delete IFF not using FIF2EB
		database.HIF	Delete IFF not sorting links
		database.H1D	Delete (Not used)
		database.STF	
FIF2EB	database.FIF		
	(FIG.IXY)		
		database.EBI	
OPTSORT	HLTSORT.CTL		
	database.HIF		
		database.HLT	

Table 9-1. Input and Output Files (cont')

Program	Input	Output	Files you can delete after build
STF2EB run	(database.EBI)		Delete, only exists if FIF2EB
links	(database.HLT)		Delete, only exists if sorted
	database.STF		Can be deleted
	database.SYS		
		*database.	
EXTWORD	database.		
	STOPWORD		
		database.EXT	
		database.HDG	
		database.MAP	
		database.OTL	Delete (not used)
FONTS2EB	database.MAP		
	database.SC	database.MAP	
		database.CFM	
OPTSORT	OPTSORT.CTL		
	database.EXT		Can be deleted
		database.SRT	
KSHEAD	database.HDG		
	database.MAP	database.MAP	
	database.SRT		Can be deleted
		database.WHD	
		*database.KSH	
		*database.HXY	
KSINDEX	database.MAP	database.MAP	
	database.WHD		Can be deleted
		*database.KSI	
OPTSORT	OPTSORT.CTL		
	database.HDG		Can be deleted
		database.HSR	
WSHEAD	database.HSR		Can be deleted
	database.MAP	database.MAP	

Table 9-1. Input and Output Files (cont')

Program	Input	Output	Files you can delete after build
		database.HHD	
		*database.WSH	
WSINDEX	database.HHD		Delete IFF not doing ENDWORD
(wsi)	database.MAP	database.MAP	
		*database.WSI	
REVERSE	database.HHD		Can be deleted
		database.REV	
OPTSORT	OPTSORT.CTL		
	database.REV	database.RHD	
WSINDEX	database.MAP	database.MAP	
(esi)	database.RHD		Can be deleted
		*database.ESI	
MSTABLE	database.MAP	database.MAP	
	database.MSx		
		database.MST	
MSREFTAB	database.Ixy		
	database.MAP	database.MAP	
		database.Rxy	
HDG2HS2	database.HDG	database.HSR	
HSR2MHD	database.HSR	database.MHD	
MHD2RMQ	database.MHD		
	database.MAP	*database.RXY	
		database.MAP	
MHD2MSI	database.MHD	database.MAP	
	database.MAP	database.MSI	
MKMENUS	database.SML	database.NIT	
		database.NTR	
MRGMENUS	database.NIT		
	database.NTS	database.SMX	
MKMAP	database.	database.MAP	

A Creating an Electronic Slide Show

This appendix tells you how to assemble a selection of graphics and display them one after another on an electronic book player, similar to displaying pictures in a "slide show."

There are five general steps to creating a slide show:

1. Create the graphics in PCX format.
2. Convert the graphics to electronic book (EB) format.
3. Create a Figure Input File (FIF file) to identify the images for the graphic search.
4. Build the electronic book file.
5. Test your slide show using the Data Discman Emulator program.

Creating the Graphics

Create your graphics. You can use any pixel editor or paint program to do this, on an Apple Macintosh or personal computer system, for example, but the pictures must end up in PCX format, with the file extension PCX, on a personal computer.

It is best to create pictures in the Data Discman screen resolution of 256 x 200 pixels in monochrome (2-color) format, because the Sony

Electronic Book Authoring System (SEBAS) tools provide minimal scaling and color conversion when the graphics are converted.

The pictures must be monochrome.

Converting the Graphics

You must convert the PCX format graphics into Sony Data Discman electronic book (EB) format graphics.

To convert your graphic images to electronic book format, you must run the PCX2EB command. PCX2EB is documented in detail in Chapter 9, SEBAS Command Reference.

The general form of the command is:

```
PCX2EB filename.PCX filename.EB
```

The first parameter is the name of the PCX file; the second parameter is the name you want to give the EB file.

Preparing a Figure Input File

A Figure Input File (FIF) is a list of the filenames of your graphic images, in a certain format. You can use a text editor to create the FIF file manually, or you can use the SGML2STF program to do it automatically.

Preparing an FIF File Automatically

The FIF file is one of the files produced when you run the SGML2STF program to create a token file from the Standard Generalized Markup Language (SGML) tagged text. If you have labelled figures in your text file, and those are the same figures you

want to display, you can use the FIF file produced by the SGML2STF program. Creating the token file is described in detail in Chapter 3, Preparing Your Text. SGML2STF is documented in detail in Chapter 9, SEBAS Command Reference.

Preparing an FIF File Manually

To create the file with a text editor, enter the filenames of your graphic images in the following format:

```
000000000 X SLIDE1.EB
000000000 X SLIDE2.EB
000000000 X SLIDE3.EB
   ...     .   ...
```

You must start in column 1, type nine zeroes, a space, an X, (not case sensitive) a space, and, starting in column 13, the filename of the graphic image.

Building the Slide Show

You must run the FIF2EB program to build the slides into a format the Sony Data Discman can use. FIF2EB is documented in detail in Chapter 9, SEBAS Command Reference.

You can use the following general form of the FIF2EB command:

```
FIF2EB filename.FIF filename.EBI
```

where:

- *filename*.FIF is the name of the Figure Input File
- *filename*.EBI is the name to give the slideshow (electronic book image) file

Although it is not required, we recommend you use the file extension EBI for the electronic book image file.

You can use the Sony Data Discman Emulator program to test the slideshow file.

Testing Your Slide Show

Run the DDMAN Data Discman Emulator program to test the appearance and sequence of your slides. The DDMAN program is documented in detail in Chapter 7, Testing Your Electronic Book.

You can invoke the emulator with the slide show by typing the following command at the system prompt:

```
DDMAN filename.EBI
```

where *filename*.EBI is the name of the image file produced by the FIF2EB program.

B Limitations and Issues

This appendix describes some issues and considerations to keep in mind when creating an electronic book.

Results of the Keyword Search

When an electronic book player displays the text resulting from a keyword search, it displays the top of the text block containing the word. The user must page down through the documents to find the sought after word. Therefore, it is best to use small text blocks in your electronic book.

If the word entered in the search EXACTLY matches a word in the text, the electronic book player displays the text containing the word.

If the word entered in the search DOES NOT EXACTLY MATCH a word that is in the text, the results of the search will be for the first word in the text that matches the characters entered.

Consider the following examples.

The index contains the words:

```
ASTONISHING
BALLFIELD
BALLPARK
CHALLENGE
...
```

The user requests a search for the word BALLGAME.
The result will be "Search unsuccessful."

The user requests a search for the word BALLPARK.
The result will be the entries containing the word BALLPARK.

The user requests a search for the word BALL.
The result will be the entries containing the word BALLFIELD, because it is the first word that matches the word BALL.

NOTE: The user is not informed that the electronic book player is showing entries for BALLFIELD when the search was for the word BALL.

C Creating and Using Special Characters

This appendix shows you how to create your own special bit-mapped characters to display on an electronic book player.

In the following steps, *database* is the name of the database title that you are creating for an electronic book player.

In general, to create and use a special character:

1. Define a bit pattern and a tag for the special character in an ASCII (text) file named *database*.SC.

2. Place the tag in your SGML text file (*database*.SML) where you want to display the special character.

3. Edit the BUILD.BAT file to create a font table and include it in the final *database*.BLD file.

Defining a Special Character

The special characters and the tags you assign to them must be in a text file with the filename of your database and the extension SC.

To define a special character:

1. Create a text (ASCII) file with the name of your database and the extension SC.

2. Create the bit pattern for the character with 0s and 1s. The 1s represent the pixels in the shape of the character. Use a grid that is 8 bits wide and 16 bits long. Refer to the examples on the following page.

3. Give the pattern a tag in the following format:

   {SCxx}

   where xx is a hexadecimal number in the range 20 to EF that identifies your character.

4. Place the tag at the beginning of the pattern for the special character, as shown in the two examples on the next page.

The following examples show the bit patterns for two special characters. Draw a line through the 1s to see the shape of the special characters. This example contains two characters defined by the tags {SC20} and {SC21} respectively. {SC20} produces an upside-down e. {SC21} produces an a with a bar over it.

```
{SC20}
00000000
00000000
00000000
00000000
00000000
00111100
01000110
00000110
00000110
01111110
01000110
01000110
00111100
00000000
00000000
00000000
{SC21}
00000000
00000000
00000000
01111110
01111110
00000000
00000000
00111100
01000110
00000110
00111110
01100110
01000110
01000110
00111110
00000000
```

Tagging Your SML File

After you give the pattern a tag, you include the tag in your SGML tagged file at the place where you want to display the special character.

This example shows a special character tag as it might appear in an SGML tagged file.

```
<h1>A
<p>{SC21}
<p>1.  Commonly found as the first entry in any
dictionary.
<p>2.  One of anything.
```

Creating Your Font Table

You must use the FONTS2EB program to create a font table for your *database*.SC file.

To create a font table, you must edit two command lines in the BUILD.BAT file created when you run the MKBATCH program.

1. Include the following statement after the EXTWORD command line:

   `fonts2eb` *`database`*

2. Add the font table to the COPY command line that concatenates all of your files into the final BLD file, like this:

   `copy` *`database`*`.+`*`database`*`.CFM+...`

The STF2EB program automatically converts your `{SCxx}` tags into the special characters they represent.

STF2EB contains options that you might like to use. The FONTS2EB and STF2EB programs are documented in detail in Chapter 9, SEBAS Command Reference.

D ISO 8859-1 Input Characters

This appendix shows the characters in the ISO 8859-1 character set that you can input from the keyboard of an electronic book player. The table also shows the ISO 8859-1 hexadecimal code for each character.

Table D-1. ISO 8859-1 Input Characters

Input Character	ISO 8859 Code
space	20H
-	2D
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39
A	41
B	42
C	43

Table D-1. ISO 8859-1 Input Characters (cont')

Input Character	ISO 8859-1 Code
D	46H
E	45
F	46
G	47
H	48H
I	49
J	4A
K	4B
L	4C
M	4D
N	4E
O	4F
P	50
Q	51
R	52
S	53
T	54
U	55
V	56
W	57
X	58
Y	59
Z	5A

Table D-1. ISO 8859-1 Input Characters (cont')

Input Character	ISO 8859-1 Code
Ä	C4H
Å	C5
Æ	C6
Ç	C7
Ë	CB
Ñ	D1
Ö	D6
Ø	D8
Ü	DC
ß	DF

E Getting Technical Support

If you have any difficulties using the Sony Electronic Book Authoring System software, and you have exhausted all attempts to find an answer to your problem by reading this guide, please call us for technical support. Each time we help someone with a problem it gives us information that we need to improve our products. You can count on our willingness to help.

Our telephone number is: (408) 372-2812.

Index

A

About This Disc, 9-32
ADDSBENT, 2-5, 9-3

B

BLANK.EB, 2-9, 2-10
BLANK.PCX, 2-9, 2-10
BUILD.BAT, 3-12, 6-2
  functions, 6-12
  running the program, 6-12
BUILDALL.BAT, 2-5
BUILDSML.BAT, 2-5

C

CATALOG file, 8-2, 9-7, 9-32
CD-ROM XA
  CDISO-DM.EXE, 8-6
  CDWO-DM.EXE, 8-6
  format, 8-1
  image file, 8-3
  premastering and mastering tools, 8-2
  sound quality levels, 5-2
  VOLUME.INF, 8-6
CDISO-DM, 2-8, 9-4
CDWO-DM, 2-8, 9-5
CHKSUB, 2-8, 9-6
consultation link, 9-33
  graphic to graphic, 4-4
Consultation link *continued*
  graphic to text, 4-5
  inserting, 3-24
  linking to audio, 3-26
  linking to graphics, 3-25
  linking to text, 3-24
  SGML tag, 4-5
  text to audio, 5-10
  text to graphic, 4-5
consultation search, 3-24
converting images, 4-2
  EB2PCX, 4-4
  PCX image, 4-3
  PCX2EB, 4-2
  SHOWEBI, 4-4
  VISTA image, 4-3
  VISTA2EB, 4-2
cue file, 5-8, 5-9, 5-10

D

Data Discman
  screen dimensions, 1-1
data files
  preparing, 3-1
DDMAN, 2-5, 2-12, 9-7
  emulator keys, 7-3
  starting, 7-1
  test database and audio file, 7-2
  testing slide show, A-4
DDMAN.EB, 2-5, 2-9, 2-10

DECODE
recording audio, 5-7
default menu language
EBIDENT program, 8-8
setting, 8-8
deletable files, 9-69
directory structure
setting up, 8-2
testing, 8-5
DMPPCX, 2-5, 4-2, 9-9
PCX graphics, 4-1
DMPVISTA, 2-5, 4-2, 9-11
VISTA graphics, 4-1
DTD file, 3-6

E

EB2PCX, 2-6, 9-12
EBI file, 9-43
EBIDENT, 2-6, 9-13
setting the default menu language, 8-3
electronic book image file, 9-19
electronic slide show
converting graphics, A-2
creating, A-1
creating graphics, A-1
FIF2EB, A-3
testing, A-4
EMPTY.DTD, 2-6
EMPTY.REF, 2-6
ENCODE
"mark" feature, 5-7
endword search, 1-2, 3-3
tags, 3-6
EXTWORD, 2-6, 9-15

F

FHFIXUP, 2-6, 9-17
FIF2EB, 2-6, 9-19, A-3
figure input file, 9-19
preparing automatically, A-2
preparing manually, A-3
font table
creating, C-4
FONTS2EB, 2-6, 9-20, C-4
formatting text, 3-20
literal text tag, 3-20

G

graphic formats
PCX file, 4-1
TGA file, 4-1
graphic search, 1-2
graphics
converting, A-2
creating, A-1
graphics consultation link
X and Y coordinates, 4-4

H

HDG2HSR, 2-6, 9-21
HLTSORT.CTL, 2-6, 2-9, 2-10
HSR2MHD, 2-6, 9-22

I

IMAGE.PCX, 2-9
IMAGINE.EB, 2-9, 2-10
IMAGINE.PCX, 2-11
input and output files, 9-69

interleaving
audio files, 5-4
INTERLV, 5-8
MKSOUND, 5-8
SINTERLV, 5-8
sound files, 5-8
INTERLV, 2-8, 9-26
creating sound file, 5-9
interleaving, 5-8
ISO 8859-1 character set, 3-22, D-1
special characters, 3-22
ISO tag
special characters, 3-23
ISO8859.ALT, 2-6

K

keyword search, 1-2, 3-3, 3-13
index file, 9-28
limitations, B-1
STOPWORD file, 3-3, 3-13
KSHEAD, 2-6, 9-27
KSINDEX, 2-6, 9-28

L

LANGUAGE, 2-6
LANGUAGE file, 8-3
link information file
creating, 4-5
format, 4-6
graphic to audio, 5-13
text to audio, 5-10
linking graphic to audio, 5-12

M

mastering, 8-2
CD-ROM XA, 8-6
menu search, 1-2, 3-3
index file, 9-29
MHD2MSI, 2-6, 9-29
MHD2RMQ, 2-6, 9-30
MHD2RMQ.CTL, 2-6, 2-9
MKBATCH, 2-6, 3-12, 9-31
automatic reference table option, 3-19
BUILD.BAT, 6-1
graphic search options, 6-8
keyword search options, 6-7
main text block options, 6-5
menu search options, 6-9
multi search options, 6-10
questions, 6-3
MKCATLOG, 2-6, 9-32
catalog file, 8-3
setting up directory structure, 8-2
MKEMENUS, 2-6, 3-12, 9-33
MKMAP, 2-6, 9-34
MKSOUND, 2-8, 9-35
interleaving, 5-8
master cue file, 5-10
MRGMENUS, 2-6, 3-12, 9-36
MSFIXUP, 2-7, 9-38
MSREFTAB, 2-7, 9-39
MSTABLE, 2-7, 9-40
multi search, 1-2, 3-15
index file, 9-30
invisible field, 3-16
question files, 3-15
reference table, 3-17
reference tables, 9-39
table of questions, 9-40
tables, 3-3
visible field, 3-16

N

NULLFILE, 2-7, 9-41

O

OPTSORT, 2-7, 9-42
OPTSORT.CTL, 2-7, 2-9, 2-11

P

PCX file, 9-9, 9-12
PCX graphics
  DMPPCX, 4-2
PCX image, 4-3
PCX2EB, 2-7, 9-43
PCYEDIT, 2-8, 9-44
PCYHDR, 2-8, 9-46
PCYLEN, 2-8, 9-47
PCYPLAY, 2-8, 9-48
  recording audio, 5-7
PCYREC, 2-8, 9-49
PCYSPLIT, 2-8, 9-50
PLAYER, 2-8, 9-51
PLAYXA, 2-8, 9-53
  check audio timings, 5-12
premastering, 8-1
  CD-ROM XA, 8-6
PUBLIC.DTD, 2-9, 2-11
PUBLIC.REF, 2-9, 2-11, 3-21
PURGE.BAT, 2-7

Q

question files
  making, 3-15
  reference table, 3-16

R

RECORDER, 2-8, 9-55
recording audio
  DECODE, 5-7
  ENCODE, 5-6
  PCYPLAY, 5-7
Red Book audio format, 5-2
reference table
  creating automatically, 3-19
  creating manually, 3-17
RERECORD, 2-8, 9-57
REVERSE, 2-7, 9-58
RPLSBENT, 2-7, 9-59

S

sample electronic books
  DDMAN, 2-12
  how to use, 2-12
  TEST.BLD, 2-11
  using, 2-11
search and retrieval function, 1-2, 3-2
  consultation search feature, 3-4
  endword search, 1-2
  graphic search, 1-2
  keyword search, 1-2
  menu search, 1-2
  multi search, 1-2
  word search, 1-2
SEBAS tools, 1-3
  hardware requirements, 2-1
  installation, 2-2, 2-3
  software requirements, 2-2
SEBAS.DTD, 2-7
SEBAS.EB, 2-9, 2-11
SEBAS.FIF, 2-9, 2-11
SEBAS.REF, 2-7

SEBAS.SYS, 2-7
SEBAS\BIN directory, 2-5
SEBAS\EBXA directory, 2-8
SEBAS\TESTALL directory, 2-9
SEBAS\TESTSML directory, 2-10
setting up directory structure
  START, 8-5
SGML, 1-3, 1-4
  DTD file, 3-6
  PUBLIC.REF file, 3-21
  substitute characters, 3-21
  tag summary, 3-4
  tagging text, 3-4
  tags, 3-3
SGML tagged file
  special characters, C-4
SGML2STF, 2-7, 9-60
  syntax errors, 3-27
SHOWEBI, 2-7, 9-62
SINTERLV, 2-9, 9-63
  interleaving, 5-8
SOUND file, 5-8
  creating, 5-1
special characters, 3-22, 9-20
  creating, C-1
  creating font table, C-4
  defining, C-1
  FONTS2EB, C-4
  ISO tag, 3-23
  SGML tagged file, C-4
SSF2SML, 2-7
START file, 9-7, 9-32
STF2EB, 2-7, 9-64
STOPWORD, 2-7, 2-9, 2-11
STOPWORD file
  creating, 3-13
syntax errors
  SGML2STF, 3-27

T

table of contents
  alphabetical search, 3-12
  making, 3-8
  making automatically, 3-11
  making manually, 3-9
  MKBATCH, 3-12
  MKEMENUS, 3-12
  MRGMENUS, 3-12
  tagging heading links, 3-9
  tagging text
    links, 3-10
tagging text
  headings, 3-7, 3-8
  multi search, 3-16
  SGML, 3-4
  SGML tag summary, 3-4
  special characters, 3-23
  syntax errors, 3-27
technical support, E-1
TEST.BLD, 2-9, 2-11, 2-12
TEST.DTD, 2-9, 2-11
TEST.I11, 2-10
TEST.MS1, 2-10
TEST.MST, 2-10
TEST.R11, 2-10
TEST.REF, 2-10, 2-11
TEST.SML, 2-10, 2-11
TEST.SYS, 2-10, 2-11
TESTFIG.EB, 2-10
TESTFIG.IXY, 2-10
TGA file, 9-11

V

VISTA file, 9-11
VISTA graphics
  DMPVISTA, 4-2

VISTA image, 4-3
VISTA2EB, 2-7, 9-66
VOLUME.INF, 2-9

W

word search, 1-2, 3-3
    tags, 3-6
WSHEAD, 2-7, 9-67
WSINDEX, 2-7, 9-68